

2. LINIJSKA STRUKTURA

2.1. Ispisati poruku "ZDRAVO" na ekranu.

Opis rješenja: Tekst koji treba ispisati na ekranu je unaprijed poznat. Zato se koristi samo naredba izlaza. Ovo je najjednostavniji oblik linijske strukture, ima samo izlaz. Tekst u naredbi ispisa se mora odvojiti sa delimiterima (za početak i kraj teksta)-jednostrukim navodnicima.

Opis programa: Listing programa u **PASCAL**-u mora početi sa naredbom **PROGRAM**, koja označava početak programa. Iza naredbe **PROGRAM** se mora navesti ime programa (na primjer: Ispis). Programski blok počinje naredbom **BEGIN**, a završava sa **END**. Ispis se izvodi naredbom **Writeln**.

Tabela 2.1. Ispis poruke

Tekstualni algoritam	Listing programa	Ekran
1. Ispisati "ZDRAVO"	<pre>PROGRAM PP2_1; BEGIN Writeln('ZDRAVO'); Readln; END.</pre>	ZDRAVO

Daljnji rad: Napisati program za ispis vlastitog imena.

2.2. Napisati program za ispis na ekranu:

Vrijednost
broja je 7

Opis rješenja: Ispisivanje navedene poruke se izvodi kao ispis u dva reda. Prvi red se sastoji od jedne riječi. Drugi red se sastoji od ispisa teksta i ispisa brojne vrijednosti. Ispis u drugom redu se može izvesti sa jednom ili sa dvije naredbe za ispis.

Opis programa: Prvi red će se ispisati kao što je prikazano u prethodnom zadatku. Drugi red se ispisuje korištenjem ispisa teksta i broja u istoj naredbi ispisa **Writeln**. Tekst se ispisuje pod apostrofima, a brojna vrijednost se navodi sa ili bez apostrofa.

Naredbom **write** kursor ostaje iza zadnje ispisano znaka. Sa naredbom **writeln** se ispisuje 7 u istom redu iza prethodnog teksta.

Tabela 2.2. Upiš i ispis

Tekstualni algoritam	Listing programa	Ekran
I. Ispisati u dva reda Vrijednost broja je 7	<pre>PROGRAM PP2_2; BEGIN Writeln('Vrijednost'); Writeln('broja je ', 7); Readln; END.</pre>	Vrijednost broja je 7

Daljnji rad: Ispisati prezime u jednom redu a ime u drugom redu.

2.3. Napisati program za izračunavanje aritmetičkog izraza:

$$9 + 2 (7 - 4)$$

Opis rješenja: Aritmetički izraz se može pisati u naredbi ispisa. Računar pri izvođenju programa izračunava vrijednost izraza i ispisuje rezultat na ekran. Znak množenja ispred zagrada u matematici se može, a ne mora navesti. Međutim, u računarskoj tehnici znak množenja je * i mora se navesti. Redoslijed izvođenja aritmetičkih operacija je isti kao i u matematici.

Opis programa: Naredba `write (writeln)` u `Pascal`-u omogućuje neposredno izračunavanje aritmetičkog izraza. Pri izvođenju programa računara izračunava vrijednost aritmetičkog izraza (izraz u zagradama) i ispisuje rezultat na ekran.

Tabela 2.3. Forula u ispisu

Tekstualni algoritam	Listing programa	Ekran
1. Izračunati izraz $9+2*(7-4)$	<pre>PROGRAM PP2_3; BEGIN Writeln(9+2*(7-4)); Readln; END.</pre>	15

Tabela 2.3a. Operatori u `PASCAL`-u

Operacija	PASCAL
Sabiranje	+
Oduzimanje	-
Množenje	*
Dijeljenje	/
Cjelobrojno dijeljenje	DIV
Ostatak cjelobrojnog djeljenja	MOD
Kvadrat (x^2)	SQR
Kvadratni korijen	SQRT

Daljnji rad: Napisati program za izračunavanje izraza: $3/2 - 3$ i $3/(2-3)$

2.4. Varijabla **x** u programu ima vrijednost 3. Ispisati na ekran vrijednost varijable **x** i uvećanu vrijednost **x** za 2.

Opis rješenja: Vrijedost varijable **x** ćemo posmatrati kao konstantu, a **y** izračunava računar. Programski jezik dozvoljava da se u naredbi ispisa koristi aritmetički izraz sa varijablama (**x + 2**).

Opis programa: Budući da **x** ima konstantnu vrijednost u programu, ovo pridruživanje se izvodi u dijelu **CONST** prije programskog bloka. Ispis konstante se izvodi sa naredbom **Writeln** ili **Write**.

Tabela 2.4. Konstanta

Tekstualni algoritam	Listing programa	Ekran
1. konstanta x	<pre>PROGRAM PP2_4; CONST x = 3; BEGIN</pre>	
2. uvećati x u naredbi ispisa	<pre> Writeln('x=',x,' y=' ,x+2); Readln; END.</pre>	x = 3 y = 5

Daljnji rad: Za **x = 4** izračunati: **x2 - x1/2**

2.5. Napisati program za izračunavanje kvadrata upisanog broja.

Opis rješenja: Prvo se upisuje vrijednost broja (ulaz). Izračunavanje kvadrata upisanog broja može biti i množenje broja sa samim sobom ($x * x$). Na kraju program se izvodi ispis.

Opis programa: Na početku programa se nalaze naredbe za deklarisanje tipa varijabli (dio **VAR**). Zatim se upisuje vrijednost varijable **x**. Obrada se izvodi sa **y := x * x**. Ispis vrijednosti je na kraju.

Tabela 2.5. Izračunavanje

Tekstualni algoritam	Listing programa	Ekran
1. upisati x	<pre>PROGRAM PP2_5; VAR x, y : Real; BEGIN Write('Broj '); Readln(x);</pre>	Broj 3
2. izračunati y	<pre> y := x * x;</pre>	
3. ispisati x i y	<pre> Writeln('x = ', x, 'y = ',y); Readln; END.</pre>	x = 3.000000E+00 y = 9.000000E+00

Drugo izvođenje programa:

x = 1.2000000000E+00 **y** = 1.4400000000E+00

Daljnji rad: Napisati program za izračunavanje izraza: $2x, a / 3$

2.6. Za upisani poluprečnik izračunati površinu kruga.

Opis rješenja: Površina kruga se izračunava prema poznatoj formuli: $p = \pi * r^2$, odnosno $p = \pi * r * r$. TURBO PASCAL sadrži vrijednost π . Ako u programskom jeziku nije sistemski ugrađena vrijednost broja π mora se izvesti pridruživanje vrijednosti za broj $\pi = 3.141593$ ili za preciznije izračunavnje se koristi izraz $\pi = 4 * \text{ARCTAN}(1.0)$.

Opis programa: Na početku programa su deklarirane varijable koje se koriste u programu. Sve vrijednosti će biti realne. Poslije upisivanja vrijednosti poluprečnika izvodi se izračunavnje. Na kraju je ispis vrijednosti.

Tabela 2.6. Površina kruga

Tekstualni algoritam	Listing programa	Ekran
1. upis poluprečnika r 2. izračunati $p = \pi * r * r$ 3. ispis vrijednosti r, p	<pre>PROGRAM P2_6; VAR r, pi, p: Real; BEGIN Write('Poluprecnik '); Readln(r); p := pi * sqr(r); Writeln('Za poluprecnik ', r); Writeln('povrsina kruga je ', p); Readln; END.</pre>	<pre>Poluprecnik 4 Za poluprecnik 4.0000000000E+00 povrsina kruga je 5.0265482457E+01</pre>

Daljnji rad: Napisati program za izračunavanje obima i površine:

- kvadrata
- pravougaonika
- trougla

2.7. Napisati program koji izračunava slijedeće izraze:

$$\begin{aligned}c &= x * y \\d &= x - y \\e &= c - x + y\end{aligned}$$

Opis rješenja: Za navedene izraze neophodno je upisati samo vrijednosti varijabli x i y . Vrijednost varijable c se izračunava.

Opis programa: Na početku programa su deklarirane varijable, u ovom zadatku su cjelobrojne vrijednosti (`Integer`). Poslije ispisa na ekranu, Dva broja, upisuje se vrijednost varijabli x i y (Na primjer 2 i 3). Vrijednosti varijabli c i d se izračunavaju na osnovu upisanih vrijednosti varijabli x i y . Pri izračunavanju vrijednosti varijable e koristi se i izračunata vrijednosti varijable c kao i upisane vrijednosti x i y . Poslije izračunavanja se izvodi ispis. Provjera programa je izvršena sa vrijednostima $x=2$ i $y=3$.

Tabela 2.7. Izračunavanje

Tekstualni algoritam	Listing programa	Ekran
	<code>PROGRAM PP2_7;</code>	
	<code>VAR</code>	
	<code> x, y, c, d, e : Integer;</code>	
	<code>BEGIN</code>	
	<code> Write('Dva broja ');</code>	Dva broja
1. upis x i y	<code> Readln(x, y);</code>	2 3
2. izračunati	<code> c := x * y;</code>	
$c = x * y$		
$d = x - y$	<code> d := x - y;</code>	
$e = c - x + y$	<code> e := c - x + y;</code>	
3. ispis x i y	<code> Writeln('x=', x, 'y=', y);</code>	x= 2 y= 3
3.1. ispis c, d, e	<code> Writeln('c=', c, ' d=',</code>	c= 6 d= -1
	<code> d, ' e=', e);</code>	e= 7
	<code> Readln;</code>	
	<code>END.</code>	

Daljnji rad: Izračunati vrijednosti prema izrazima:

$$\begin{aligned}a &= x * y \\b &= x + y \\c &= b / x\end{aligned}$$

2.8. Na ekranu ispisati Znak a zatima sa tastature upisati znak.

Opis rješenja: Vrijednost koja se upisuje ne mora biti numerička vrijednost. Računar zahtjeva da se koristi tip varijable(**CHAR**).Pravi se razlika razliku da li se upisuje jedan znak (**CHAR**)ili više znakova(**STRING**). Prikazan je slučaj kad se upisuje jedan znak.

Opis programa: Poslije ispisa teksta Znak (naredba `write('Znak ')`) slijedi upis. Budući da je navedeno da će se upisati samo jedan znak (`VAR x: Char;`).

Tabela 2.8. Znak

Tekstualni algoritam	Listing programa	Ekran
	<code>PROGRAM PP2_8;</code>	
	<code>VAR</code>	
	<code> x: Char;</code>	
	<code>BEGIN</code>	
	<code> Write('Znak ');</code>	Znak
1. upisati znak	<code> Readln(x);</code>	G
2. isisati znak	<code> Writeln('Znak ',x);</code>	Znak G
	<code> Readln;</code>	
	<code>END.</code>	

Daljnji rad: Upisati i ispisati znak E.

2.9. Sa tastature upisati više znakova i ispisati na ekran.

Opis rješenja: Oznaka varijable za znakove u **PASCAL**-u je **STRING**.

Opis programa: Znakovi se upisuju sve dok se ne pritisne tipka **ENTER** jer je navedeno (`VAR x: String`). Deklarisanje varijable je različito za jedan i više znakova. Bitno je deklarisanje varijable a ne ime varijable.

Tabela 2.9. String

Tekstualni algoritam	Listing programa	Ekran
	<code>PROGRAM PP2_9;</code>	
	<code>VAR</code>	
	<code> x: String;</code>	
	<code>BEGIN</code>	
	<code> Write('String ');</code>	String
1. upisati STRING	<code> Readln (x);</code>	Zdravo
2. ispisati STRING	<code> Writeln('String ',x);</code>	String Zdravo
	<code> Readln;</code>	
	<code>END.</code>	

Daljnji rad: Upisati i ispisati TESLA

3. RAZGRANATA STRUKTURA

3.1. Sa tastature upisati broj. Ako je veći od nule napisati "BROJ JE POZITIVAN"

Opis rješenja: Sa tastature se upisuje vrijednost varijable *a*. Zatim se sa logičkim izrazom $a > 0$ u naredbi **IF** izvodi poređenje da li je upisana vrijednost pozitivna. Ako je logički izraz istinit na ekranu se ispisuje "POZITIVAN", inače za ostale slučajeve nema ispisa.

Opis programa: Poslije upisa vrijednosti broja *a* (**Readln(a)**) izvodi se provjera vrijednosti varijable *a* (da li je POZITIVAN). Provjera se izvodi sa logičkim izrazom $a > 0$ u naredbi **IF**. Naredba **IF** se sastoji od logičkog izraza i obrade. Logički izraz ($a > 0$) je istinit kad je upisana vrijednost sa tastature pozitivna. Za ove vrijednosti na ekranu će se ispisati POZITIVAN. Inače (za ostale slučajeve) nema ispisa.

Tabela 3.1. Pozitivan

Tekstualni algoritam	Listing programa	Ekran
1. upis vrijednosti varijable <i>a</i> 2. ako je broj veći od nule tada, ispiši POZITIVAN	<pre>PROGRAM P3_1; VAR a: Real; BEGIN Write('Vas broj '); Readln(a); IF a > 0 THEN Writeln('POZITIVAN'); Readln; END.</pre>	Vas broj 3 POZITIVAN

Daljnji rad: Provjeriti da li je upisani broj paran.

3.2. Provjeriti da li je od dva upisana broja prvi djeljiv sa drugim.

Opis rješenja: Prvo se sa tastature upisuju dvije vrijednosti (**a**, **b**). Provjera da li je prvo upisani broj (**a**) djeljiv sa drugim se izvodi sa logičkim izrazom **a MOD b** u naredbi **IF**. Ako je logički izraz istinit ispisuje se na ekranu **DJELJIV**, inače nema ispisa. Sa naredbom **a MOD b** se izračunava ostatak dijeljenja broja **a** sa brojem **b**. Ostatak dijeljenja je jednak **0** ako je broj **a** djeljiv sa brojem **b**. Logički izraz za provjeru djeljivosti (**a MOD b = 0**) je istinit ako je **a** djeljivo sa **b** i tada se na ekranu ispisuje **DJELJIV**. Inače za ostale slučajeve nema ispisa.

Opis programa: Poslije upisa vrijednosti varijabli **a** i **b** sa tastature, provjerava se da li je **a** djeljivo sa **b**. Logički izraz u naredbi **IF** glasi **a mod b = 0**. Ako je logički izraz istinit ispisuje se **DJELJIV**.

Tabela 3.2. Provjera djeljivosti brojeva

Tekstualni algoritam	Listing programa	Ekran
1. upis vrijednosti varijabli a,b 2. ako je prvi broj djeljiv sa drugim ispiši DJELJIV	<pre>PROGRAM PP3_2; VAR a, b: Integer; BEGIN Write('Vasi brojevi'); Readln(a, b); IF a MOD b = 0 THEN Writeln('DJELJIV'); Readln; END.</pre>	<pre>Vasi brojevi 6 3 DJELJIV</pre>

Daljni rad: Sa tastature upisati riječ. Ako se upiše **BASIC** ili **Pascal** ispisat: "Ispisano je ime programskog jezika".Sa tastature upisati ime. Ako upišete **ANDRIĆ** ispisati "To je veliki pisac"

3.2. IF THEN ELSE

3.3. Sa tastature upisati broj i izračunati njegovu recipročnu vrijednost ($y = 1 / x$) i ispisati je.

Opis rješenja: Recipročna vrijednost broja je 1 podijeljen sa upisanim brojem. Međutim u slučaju da je upisani broj jednak nuli trebalo bi se dijeliti sa nulom, a dijeljenje je nemoguće. Proizilazi da u odnosu na vrijednost upisanog broja postoje dvije akcije (obrade). Izbor se izvodi na osnovu logičkog izraza $n=0$, korištenjem razgranate strukture tipa **IF THEN ELSE**. Za vrijednosti $n \neq 0$ izračunava se recipročna vrijednost (dio iza **THEN**), dok se za $n = 0$ ispisuje da je dijeljenje nemoguće.

Opis programa: Logički izraz za provjeru da li je upisani broj jednak nuli je ($n=0$). Ako je logički izraz istinit tada se ispisuje "Dijeljenje sa nulom je nemoguće". Inače se ispisuje "Recipročna vrijednost je " kao i izračunata recipročna vrijednost."

Tabela 3.3. Izračunati recipročnu vrijednost broja

Tekstualni algoritam	Listing programa	Ekran
1. upis vrijednosti broja n 2. ako je $n=0$ 2.1. ispiši "DIJELJENJE SA NULOM" inače 2.2. izračunaj i ispiši recipročnu vrijednost ($1/n$)	<pre>PROGRAM PP3_3; VAR n: Real; BEGIN Write('Vas broj '); Readln(n); IF n = 0 THEN Writeln('Dijeljenje sa nulom '); ELSE Writeln('Recipročna vrijednost je', 1/n); Readln; END.</pre>	Vas broj 0 Dijeljenje sa nulom je nemoguće

Daljnji rad: Za upisani broj provjeriti da li je paran.

3.4. Za upisani broj ispisati znak broja prema izrazu:

$$z = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

Opis rješenja: Navedeni izraz ima sljedeće značenje: ako je $x \geq 0$, tada je $z=1$, inače je $z=0$. Ovo je razgranata struktura tipa **IF THEN ELSE**.

IF THEN Obrada1 ELSE Obrada2

Elementi ove strukture su: LOGIČKI IZRAZ ($x \geq 0$), Obrada1 ($z=1$) i Obrada2($z=0$). Logički izraz je istinit ako je $x = 0$ i tada se izvodi Obrada1 ($z=1$), inače kad je logički izraz lažan izvodi se Obrada2 odnosno z prima vrijednost nule.

Opis programa: Na početku programa su navedeni tipovi varijabli. Program počinje sa tekstom da treba upisati broj. Po upisu broja izvodi se provjera vrijednosti broja (da li je negativan). Ako je upisani broj veći ili jednak nuli varijabla z postaje 1 inače z postaje 0. Na kraju je ispis vrijednosti varijabli x i z .

Tabela 3.4. Upisanom broju pridružiti odgovarajući znak

Tekstualni algoritam	Listing programa	Ekran
1. upis vrijednosti broja x 2. ako je $x \geq 0$ 2.1. tada je $z=1$, 2.2. inače $z=0$ 3. ispis vrijednosti x i z	<pre> PROGRAM PP3_4; VAR x: Real; z: Integer; BEGIN Write('Vas broj '); Readln(x); IF x >= 0 THEN z := 1 ELSE z := 0; Writeln('Broj = ', x, ' znak = ', z); Readln; END. </pre>	<pre> Vas broj 5 Broj = 5.0000000000E+00 znak = 1 </pre>

Drugo izvođenje programa:

Vas broj -2

Broj = -2.0000000000E+00 znak = 0

Daljnji rad: Na ekranu ispisati "Otkucajte nadimak pjesnika Jovana Jovanovica". Ako se otkuca Zmaj ili ZMAJ ispisati "ODLICNO", inače na druge odgovore ispisati "ISPRAVAN ODGOVOR JE ZMAJ".

3.5. Sa tastature upisati dva broja (x i a). Varijabli z pridružiti vrijednost prema slijedećem izrazu:

$$z = \begin{cases} x+a & x > a \\ x-a & x \leq a \end{cases}$$

Opis rješenja: Po upisu x i a sa tastature provjerava se odnos x i a . Ako je logički izraz $x > a$ istinit vrijednost varijable z se određuje prema izrazu $z = x + a$. Inače za ostale vrijednosti varijable x varijabla z prima vrijednosti prema izrazu $z = x - a$ (Obrada2).

Opis programa: Poslije upisa vrijednosti za varijable x i a izvodi se poređenje ovih vrijednosti. Logički izraz je $x > a$ i ako je istinit, tada z prima vrijednost $x + a$ ($z := x + a$). Inače, (kad je logički izraz lažan) z se izračunava po izrazu: $z := x - a$. Ispis svih vrijednosti je na kraju.

Tabela 3.5. Izračunavanje vrijednosti z prema izrazima

Tekstualni algoritam	Listing programa:	Izvođenje programa:
1. upis vrijednosti brojeva x i a 2. ako je $x > a$ 2.1. tada je $z = x + a$ 2.2. inače $z = x - a$ 3. Ispisati x , a , z	<pre> PROGRAM PP3_5; VAR x, a, z: Real; BEGIN Write('Dva broja '); Readln(x, a); IF x > a THEN z := x + a ELSE z := x - a; Writeln('x=', x, ' a=', a, ' z=', z); Readln; END. </pre>	Dva broja 1 3 $x = 1.0000000000E+00$ $a = 3.0000000000E+00$ $z = -2.0000000000E+00$

Daljnji rad: Upisati dva broja (x i a). Varijabli z pridružiti vrijednost prema slijedećem izrazu:

$$z = \begin{cases} x/3 & x > a \\ x+3 & x \leq a \end{cases}$$

3.3. KOMBINOVANI ZADACI

3.6. Od tri upisana broja (a, b, c) sa tastature naći najveći.

Opis rješenja: Vrijednosti tri broja se učitavaju sa tastature. Nova varijabla **max** prima vrijednost varijable **a**. Zatim poredimo da li je **b** veće od **max**, ako jeste tad **max** postaje **b**. Slijedeće poređenje je da li je **c** veće od **max**. Ako je **c** veće od **max** tad je to nova vrijednost **max**.

Opis programa: Algoritam nalaženja najvećeg broja je dat u opisu rješenja. Po upisu vrijednosti varijabli **a, b i c** postavlja se početna vrijednost varijable **max**. Logičkim izrazom **b>max** u naredbi **IF** se provjerava da li je vrijednost varijable **b** veća od pretpostavljenog maksimuma. Ako je logički izraz istinit mijenja se vrijednost varijable **max**. Isto poređenje i po potrebi zamjena se obavlja u slijedećoj naredbi **IF**. Ispis vrijednosti je na kraju.

Tabela 3.6. Od tri upisana broja naći najveći

Tekstualni algoritam	Listing programa	Ekran
1. upis tri broja a,b,c 2. max=a 3. ako je b>max tada je max=b 4. ako je c>max tada je max=c 5. ispisati a,b,c,max	<pre> PROGRAM PP3_6; VAR a, b, c, max: Real; BEGIN Write('Vasa tri broja '); Readln(a, b, c); max := a; IF b>max THEN max := b; IF c>max THEN max := c; Writeln('Od brojeva ', a, ' ', b, ' ', c); Writeln('najveci je ', max); Readln; END. </pre>	Vasa tri broja 3 2 5 Od brojeva 3.0000000000E+00 2.0000000000E+00 5.0000000000E+00 najveći je 5.0000000000E+00

Daljnji rad: Zavisno o upisanoj vrijednosti časa 0-23 ispisati poruku:

Sati	Poruke
0 - 8	Dobro jutro
9 - 18	Dobar dan
19 - 23	Dobro veče

3.7. Provjeriti da li je upisani broj pozitivan, negativan ili nula.

Opis rješenja: Zadatak ima tri ispisa, međutim može se riješiti sa dvije **IF** naredbe budući da naredba **IF THEN ELSE** sadrži dvije mogućnosti. Sa prvom se eliminiše jedno rješenje, a sa njenim **ELSE** dijelom se prelazi na odlučivanje o drugom i trećem rješenju. Izbor između ove druge dvije mogućnosti se izvodi sa novom (drugom) **IF THEN ELSE** naredbom.

Napomena: Jasno je da se zadatak može riješiti i sa tri **IF THEN** naredbe, ili sa dvije **IF THEN** naredbe koje koriste **GOTO** naredbu za skok na kraj.

Opis programa: Po upisu vrijednosti broja **a** izvodi se provjera da li je broj pozitivan. Ako je logički izraz (**A>0**) u **IF** naredbi istinit ispisuje se **POZITIVAN**. Inače se prelazi na dio naredbe **ELSE** unutar koje je druga **IF** naredba sa logičkim izrazom **a<0**. Ako je ovaj logički izraz istinit ispisuje se **NEGATIVAN**. Inače se ispisuje **NULA**. Ispred **ELSE** naredbe se nikad ne piše tačka zaraz (;) jer je to dio **IF THEN ELSE** naredbe.

Tabela 3.7. Provjera upisanog broja(pozitivan,negativan ili nula)

Tekstualni algoritam	Listing programa	Ekran
1. upis broja	<code>PROGRAM PP3__7; VAR a : Real; BEGIN Write('Vas broj '); Readln(a);</code>	Vas broj 3
2. ako je a>0	<code> IF a > 0 THEN</code>	POZITIVAN
2.1. tada ispiši POZITIVAN	<code> Writeln('POZITIVAN')</code>	
inače	<code> ELSE</code>	
2.2., ako je a<0	<code> IF a < 0 THEN</code>	
2.2.1. tada ispiši NEGATIVAN	<code> Writeln('NEGATIVAN')</code>	
	<code> ELSE</code>	
2.2.2. inače ispiši NULA	<code> Writeln('NULA'); Readln;</code>	
	<code>END.</code>	

Drugo izvođenje programa

Vas broj -7
NEGATIVAN

Daljnji rad: Napisati program za izračunavanje vrijednosti **z** prema izrazu:

$$y = \begin{cases} b / a, & a > 0 \\ a + 3, & -3 < a \leq 0 \\ a + b, & a \leq -3 \end{cases}$$

3.8. Napisati program za izračunavanje vrijednosti z prema izrazu:

$$y = \begin{cases} b/a, & a > 0 \\ a+3, & -3 < a \leq 0 \\ a+b, & a \leq -3 \end{cases}$$

Opis rješenja: Zadatak ima tri različita izraza za izračunavanje varijable y . Izložena je jedna od mogućih varijanti rješenja. Koristiće se dvije **IF THEN ELSE** naredbe i to tako da je druga **IF** naredba u **ELSE** dijelu prve naredbe. Prva **IF** naredba u dijelu **THEN** rješava jedna mogućnost, a sa druga **IF** naredba preostale dvije mogućnosti. Jasno je da se zadatak može riješiti i sa tri **IF THEN** naredbe, ili sa dvije **IF THEN** naredbe koje koriste **GOTO** naredbu za skok na kraj.

Opis programa: Logičkim izrazom $a > 0$ u prvoj **IF** naredbi se provjerava da li je broj pozitivan. Ako je ovaj logički izraz istinit (a veće od 0) varijabla y prima vrijednost prema izrazu $y := b/a$. Inače se sa drugom **IF** naredbom provjerava da li je vrijednost varijable a u intervalu $-3 < a \leq 0$. Provjera ovog intervala se izvodi sa logičkim izrazom koji mora koristiti naredbu **AND**, budući da se varijabla može porediti samo sa jednim logičkim operatorom. Ako je ovaj logički izraz ($-3 < a$ **AND** $a \leq 0$) istinit varijabli y se pridružuje vrijednost prema izrazu: $y := a+3$. Inače, ako je izraz lažan pridruživanje se izvodi sa izrazom $y := a+b$.

Tabela 3.8. Izračunavanje vrijednosti z prema izrazima

Tekstualni algoritam	Listing programa	Ekran
<ol style="list-style-type: none"> 1. upis varijable a, b 2. ako je $a > 0$ <ol style="list-style-type: none"> 2.1. tada je $y = b/a$ inače, 2.2. ako je $-3 < a$ AND $a \leq 0$ <ol style="list-style-type: none"> 2.2.1. tada je $y = a+3$ inače je 2.2.2. $y = a+b$ 3. ispiši a, b, y 	<pre>PROGRAM PP3_8; VAR a, b, y : Real; BEGIN Write('Dva broja '); Readln(a, b); IF a>0 THEN y:=b/a ELSE IF (-3<a) AND (a<=0) THEN y:=a+3 ELSE y:=a+b; Writeln('a=', a, ' b=', b, ' y=', y); Readln; END.</pre>	<p>Dva broja 3 6</p> <p>a=3 b=6 y= 2</p>

Daljnji rad: Ispred **ELSE** naredbe se nikad ne piše tačka zaraz (;) jer je to dio **IF THEN ELSE** naredbe.

4. CIKLIČNA STRUKTURA

4.1. Ispis prvih n prirodnih brojeva FOR petljom.

Opis rješenja: Izvođenje programa počinje učitavanjem vrijednosti n do koje se izvodi ispis. FOR petlja omogućuje promjenu vrijednosti kontrolne varijable od početne do krajnje vrijednosti. Kontrolna varijabla za ovaj zadatak je i . Početna vrijednost je 1, a krajnja n . Vrijednost varijable i se ispisuje na ekran pri svakom prolazu kroz petlju.

Opis programa: Po učitavanju broja do kojeg se izvodi ispis prelazi se na FOR naredbu. Ova naredba postavlja vrijednost kontrolne varijable i u jedinicu. Ispis vrijednosti varijable i je u slijedećoj naredbi. Poslije ispisa računar prenosi izvođenje programa na naredbu FOR koja uvećava vrijednost kontrolne varijable i te je poredi sa krajnjom vrijednošću. Budući da nije postignuta krajnja vrijednost ispisuje se vrijednost varijable i (sad je to 2). Petlja se ponavlja sve dok kontrolna varijable ne primi vrijednost veću od krajnje vrijednosti.

Naredba FOR sadrži jednu naredbu, ako treba izvesti više od jedne naredbe u FOR petlji te se naredbe svode na jednu korištenjem naredbi BEGIN i END.

Tabela 4.1. Ispis prvih n prirodnih brojeva sa FOR petljom

Tekstualni algoritam	Listing programa:	Ekran
1. učitati do kog broja (n) 2. za $i=1$ do n radi 2.1. ispisati vrijednost i 2.2. idi na 2	<pre>PROGRAM PP4_1; VAR i, n: Integer; BEGIN Write('Do broja '); Readln(n); FOR i := 1 TO n DO Writeln('i = ', i); Readln; END.</pre>	Do broja 3 i = 1 i = 2 i = 3

Daljnji rad: Ispis prvih n prirodnih brojeva unazad. Riješiti sa FOR petljom.

4.2. Izračunati sumu prvih n prirodnih brojeva. Koristiti FOR petlju.

Opis rješenja: Početna vrijednost sume s (0). FOR petlja obezbjeđuje promjenu vrijednosti kontrolne varijable i od jedan (1) do n . Formiranje sume se izvodi unutar FOR petlje. Izraz $s:=s+i$ je računarski, a ne matematički izraz. Njim se predstavlja dinamika promjene vrijednosti varijable s . Desni dio izraza $(s+i)$ predstavljala uvećanje vrijednosti varijable s za vrijednost kontrolne varijable i . Izračunata vrijednost se pridružuje varijabli s . Zato se izraz $s:=s+i$ čita s i postaje $s+i$ tj. s prima vrijednost $s+i$. Postupak se ponavlja sve dok je $i \leq n$.

Opis programa: Po učitavanju broja n do kog se izvodi sabiranje postavlja se početna vrijednost sume s (0). Sa naredbom FOR je formirana petlja koja mijenja vrijednosti kontrolne varijable i od jedan (1) do n . Formiranje sume s je obezbjeđeno unutar ove petlje. Pri prvom prolazu kroz petlju se na vrijednost sume (nula) dodaje vrijednost varijable i (jedan) ($s+i=1+0=1$). Izračunata vrijednost se pridružuje varijabli s ($s:=1$). Postupak se ponavlja za $i=2$ ($s+i=2+1=3$) i tako redom dok i ne postane veće od n . Na kraju je ispis izračunate sume s .

Tabela 4.2. Sumu prvih n prirodnih brojeva - FOR petlja

Tekstualni algoritam	Listing programa:	Ekran
1. učitati broj (n) 2. početna vrijednost sume ($s=0$) 3. za $i=1$ do n radi 3.1. formirati novu sumu ($s=s+i$) 4. ispisati izračunatu sumu s	<pre>PROGRAM PP4_2; VAR i, n, s: Integer; BEGIN Write('Do broja n '); Readln(n); s := 0; FOR i := 1 TO n DO s := s + i; Writeln('Suma do ', n, ' je ', s); Readln; END.</pre>	Do broja 3 Suma do 3 je 6

Daljnji rad: Naći proizvod prvih n prirodnih brojeva, koristiti FOR petlju.

4.3. Naći proizvod prirodnih brojeva u intervalu od k do n , koristiti FOR petlju.

Opis rješenja: Početna vrijednost proizvoda je $p:=1$. Izraz za izračunavanja proizvoda je $p:=p*i$. Tumačenje ovog izraza je slično objašnjenju izraza za sabiranje ($s:=s+i$). Desni dio izraza ($p*i$) predstavlja proizvod vrijednosti varijabli p i i . Dobiveni proizvod se pridružuje varijabli p . Jasno je da postoji razlika između ovog i matematičkog tumačenja navedenog izraza. Inače izraz se čita kao p postaje $p*i$ ili kao p prima vrijednost $p*i$.

Opis programa: Na početku programa se učitavaju granice intervala i postavlja početna vrijednost proizvoda ($p:=1$). FOR petlja obezbjeđuje promjenu vrijednosti kontrolne varijable i od k do n . Pri svakom prolazu kroz petlju vrijednost proizvoda p se množi sa vrijednošću varijable i ($p*i$). Ova izračunata vrijednost se pridružuje varijabli p . Petlja se ponavlja sve dok vrijednost kontrolne varijable i ne bude veća od n . Ispis se obavlja na kraju programa.

Tabela 4.3. proizvod brojeva od k do n - FOR petlja.

Tekstualni algoritam	Listing programa:	Ekran
<ol style="list-style-type: none"> 1. učitati granice intervala (k, n) 2. ako su pogrešne granice, idi na 1 ($k>n$) 3. postaviti početnu vrijednost proizvoda ($p=1$) 4. za $i=k$ do n radi <ol style="list-style-type: none"> 4.1. formirati novi proizvod ($p=p*i$) 5. ispisati izračunati proizvod 	<pre> PROGRAM PP4_3; VAR i, k, n, p: Integer; BEGIN REPEAT Write('Granice k n '); Readln(k, n); UNTIL k<= n; p := 1; FOR i := k TO n DO p := p * i; Writeln('Od ', k, ' do ', n, ' proizvod je ', p); Readln; END. </pre>	<p>Granice k n 5 7</p> <p>Od 5 do 7 proizvod je 210</p>

Daljnji rad: Izračunati sumu i proizvod prirodnih brojeva od k do n . Koristiti jednu i dvije FOR petlju.

4.4. Izračunati sumu parnih prirodnih brojeva u intervalu od k do n .

Opis rješenja: Ovaj program se odlikuje time da u FOR petlji ima provjeru da li je kontrolna varijabla djeljiva sa 2 tj. da li je parna. Provjera se izvodi sa logičkim izrazom $i \text{ MOD } 2 = 0$ u naredbi IF. Sabiraju se samo brojevi koji zadovoljavaju dati logički izraz tj. kad je logički izraz istinit.

Opis programa: Po upisu granice intervala i postavlja se početna vrijednost sume ($s:=0$). Zatim počinje FOR petlja. Prva vrijednost kontrolne varijable i je k . Naredba IF sadrži logički izraz $i \text{ MOD } 2 = 0$. Ako je ovaj logički izraz tačan tada se uvećava suma za vrijednost kontrolne varijable i . Proces se ponavlja dok vrijednost kontrolne varijable ne bude veća od n . Ispis je na kraju.

Tabela 4.4. Suma parnih brojeva od k do n

Tekstualni algoritam	Listing programa:	Ekran
1. učitati granice intervala (k, n)	<pre>PROGRAM PP4_4; VAR i, k, n, s: Integer; BEGIN REPEAT Write('Granice k n '); Readln(k, n);</pre>	Granice k n 3 10
3. postaviti početnu vrijednost sume	<pre>UNTIL k<= n; s := 0;</pre>	
4. za $i=k$ do n radi	<pre>FOR i := k TO n DO</pre>	
4.1. ako je i parno tada	<pre> IF i MOD 2 = 0</pre>	
formirati sumu ($s=s+i$)	<pre> THEN s := s + i;</pre>	
5. ispisati izračunatu sumu	<pre> Writeln('Od ', k, ' do ', n, ' suma parnih je ', s); Readln; END.</pre>	Od 3 do 10 suma parnih je 28

Daljnji rad: Izračunati sumu neparnih prirodnih brojeva u intervalu od k do n .

4.2. WHILE petlja

4.5. Ispis prvih N prirodnih brojeva sa WHILE petljom.

Opis rješenja: Kontrolnoj varijabli *i* se pridjeljuje vrijednost jedan (1). **WHILE** petlja se ponavlja sve dok je logički izraz $i \leq n$ istinit. Naredbom $i = i + 1$ se uvećava vrijednost varijable *i* za jedan. Računar ovu operaciju izvodi na slijedeći način. Prvo vrijednosti varijable *i* dodaje 1. Zatim ovu vrijednost pridružuje varijabli *i*. Izvršenje programa prelazi na početak **WHILE** petlje, koja obezbjeđuje ponavljanje.

Opis programa: Poslije upisa vrijednosti *n* (koliko će se brojeva ispisati) varijabli *i* se pridružuje početna vrijednost ($i = 1$). Petlja počinje naredbom **WHILE**. Logički izraz u **WHILE** naredbi obezbjeđuje ponavljanje sve dok je logički izraz ($i \leq n$) istinit. Vrijednost varijable *i* se uvaćava za 1 korištenjem izraza $i = i + 1$. Računar prvo izvodi desni dio ovog izraza, uvećanje vrijednosti varijable *i* za jedan ($i + 1$). Izračunatu vrijednost računara pridružuje varijabli *i*. Prelazi se na početak petlje, tj ponavlja se petlja dok je istinit logički izraz u **WHILE** naredbi.

Tabela 4.5. Ispis pomoću **WHILE** petlje

Tekstualni algoritam	Listing programa:	Ekran
<ol style="list-style-type: none"> 1. upis do broja (<i>n</i>) 2. postaviti početnu vrijednost ($i = 1$) 3. dok je $i \leq n$ radi <ol style="list-style-type: none"> 3.1. ispisati vrijednost <i>i</i> 3.2. uvećati <i>i</i> 3.3. idi na 3 	<pre>PROGRAM PP4_5; VAR i, n: Integer; BEGIN Write('Do broja '); Readln(n); i := 1; WHILE i <= n DO BEGIN Writeln('i = ', i); i := i + 1; END; Readln; END.</pre>	<p>Do broja 3</p> <p>i = 1 i = 2 i = 3</p>

Daljnji rad: Ispisati prvih *n* prirodnih brojeva unazad koristeći **WHILE** petlju.

4.6. Naći sumu prvih n prirodnih brojeva. Zadatak riješiti sa **WHILE** petljom.

Opis rješenja: Početna vrijednost varijable s (suma) je 0. Suma se formira po izrazu $s:=s+i$. Računar pri izvođenju prvo izvodi desni dio, sabira vrijednost s i vrijednost i . Izračunata vrijednost se pridružuje varijabli s . Zatim se vrijednost varijable i uvećava za jedan ($i=i+1$). Postupak uvećanja sume i vrijednosti varijable i se ponavlja sve dok je vrijednost varijable i manja ili jednaka n .

Opis programa: Početne vrijednosti varijable s ($s:=0$) i varijable i ($i=1$). Logički izraz $i<=n$ obezbjeđuje ponavljanje **WHILE** petlje. Izrazom $s:=s+i$ se izvodi formiranje sume. Po izvedenom sabiranju ($s+i$) izračunata vrijednost se pridružuje sumi s . Zatim se uvećava vrijednost varijable i ($i:=i+1$). Izračunata vrijednost sume s se ispisuje na kraju programa.

Tabela 4.6. Suma prvih n prirodnih brojeva

Tekstualni algoritam	Listing programa:	Ekran
<p>1. upis do broja (n) 2. početne vrijednosti $s=0$, $i=1$ 3. dok je $i<=n$ radi 3.1. formirati novu sumu 3.2. uvećati kontrolnu varijablu i za 1 4. ispisati izračunatu sumu</p>	<pre>PROGRAM PP4_6; VAR i, n, s: Integer; BEGIN Write('Do broja '); Readln(n); s := 0 ; i := 1; WHILE i<=n DO BEGIN s := s + i; i := i + 1; END; Writeln('Suma prirodnih brojeva do ', n, ' je ', s); Readln; END.</pre>	<p>Do broja 3</p> <p>Suma prirodnih brojeva do 3 je 6</p>

Daljnji rad: Izračunati proizvod prvih N prirodnih brojeva.

4.8. Prebrojati prirodne brojeve iz intervala od k do n koji su djelivi sa 3. Za generisanje prirodnih brojeva koristiti **WHILE** petlju.

Opis rješenja: Izbor djeljivih sa 3 se izvodi logičkim izrazom $i \text{ MOD } 3=0$. Logički izraz je istinit kad pri dijeljenju broja sa 3 nema ostatka ($\text{ostatak}=0$), formira se nova vrijednost brojača. Početna vrijednost brojača je nula ($br:=0$). Vrijednost brojača se uvačava za jedan ($br+1$) kad je logički izraz $i \text{ MOD } 3=0$ istinit.

Opis programa: Početne vrijednosti brojača $br:=0$ i kontrolne varijable $i:=k$. Provjera djeljivosti broja sa 3 se izvodi logičkim izrazom $i \text{ MOD } 3=0$. Logički izraz je istinit ako pri dijeljenju nema ostatka. Brojač br se uvećava za jedan ($br+1$) za svaki broj koji je djeljiv sa 3. Vrijednost se pridružuje varijabli br . Petlja se ponavlja za sve vrijednosti upisanog intervala (k, n). Ispis izračunate vrijednosti brojača je na kraju.

Tabela 4.8. Prebrojati prirodne brojeve od k do n djelive sa 3

Tekstualni algoritam	Listing programa:	Ekran
<p>1. granice intervala (k, n)</p> <p>2. ako su pogrešne granice ($k>n$), idi na 1</p> <p>3. postaviti početne vrijednosti brojača i kontrolne varijable i</p> <p>4. dok je $i \leq n$ radi</p> <p>4.1. ako je broj i djeljiv sa 3 uvećati brojač</p> <p>4.2. uvećati kontrolnu varijablu i</p> <p>5. ispis izračunatu vrijednosti brojača</p>	<pre>PROGRAM PP4_8; VAR i, k, n, br: Integer; BEGIN REPEAT Write('Granice intervala '); Readln(k, n); UNTIL k<n; br := 0 ; i := k; WHILE i <= n DO BEGIN IF i MOD 3 = 0 THEN br := br + 1; i := i + 1; END; (* while *) Writeln('Od ', k, ' do ', n, ' ima ', br, ' brojeva djeljivih sa 3'); Readln; END.</pre>	<p>Granice 2 16</p> <p>Od 2 do 16 ima 5 brojeva djeljivih sa 3</p>

Daljnji rad: Izračunati sumu neparnih i proizvod parnih brojeva u intervalu od k od n sa dvije **WHILE** petlje, a drugoj verziji sa jednom petljom.

4.9. Ispis prvih n prirodnih brojeva. Zadatak riješiti **REPEAT** petljom.

Opis rješenja: Sa tastature se upisuje vrijednosti varijable n do koje se izvodi ispis. Kontrolnoj varijabli i se pridružuje vrijednost jedan i ispisuje na ekran. Uvećava se vrijednost varijable i za $(i+1=1+1=2)$. Vrijednost se ispisuje na ekran. Ponovo se uvećava vrijednost kontrolne varijable i za jedan $(i+1=2+1=3)$. Poslije ispisa kontrolne varijable ponavlja se uvećanje i i ispis, sve dok kontrolna varijabla i ne primi vrijednost veću od n .

Opis programa: Poslije upisa vrijednosti n (koliko će se brojeva ispisati) postavlja se vrijednost varijable i na početnu vrijednost ($i:=1$). Petlja počinje sa naredbom **REPEAT**. Slijedi ispis vrijednosti varijable i . Zatim se vrijednost varijable i uvećava za 1. Naredbom **UNTIL** se provjerava da li je vrijednost varijable i veća od n . Ponavljanje naredbi u petlji je sve dok logički izraz u naredbi **UNTIL** ne bude istinit.

Tabela 4.9. Ispis prvih n prirodnih brojeva

Tekstualni algoritam	Listing programa:	Ekran
1. upis do broja (n) 2. postaviti početnu vrijedost ($i=1$) 3. ispisati vrijednost i 4. uvećati i ($i=i+1$) 5. kraj petlje	<pre>PROGRAM PP4_9; VAR i, n: Integer; BEGIN Write('Do broja '); Readln(n); i := 1; REPEAT Writeln('i = ',i); i := i + 1; UNTIL i > n; Readln; END.</pre>	<pre>Do broja 3 i = 1 i = 3 i = 2</pre>

Daljnji rad: Ispisati prvih n prirodnih brojeva unazad.

4.10. Naći sumu prvih n prirodnih brojeva. Zadatak riješiti **REPEAT** petljom.

Opis rješenja: Početna vrijednost varijable s (suma) je 0, a kontrolne varijable i jedan (1). Vrijednost sume s se uvećava za vrijednost kontrolne varijable i ($s+i=0+1=1$). Izračunata vrijednost se pridružuje varijabli s sa izrazom ($s=s+i$) (odnosno $s:=s+i$). Vrijednost kontrolne varijable i se uvećava za 1 te se dodaje na vrijednost sume s ($s+i=1+2=3$). Ovo je nova vrijednost sume s . Proces uvećanja kontrolne varijable i i dodavanja na vrijednost sume se ponavlja sve dok je vrijednost kontrolne varijable i manja ili jednaka n .

Opis programa: Vrijednost varijable n (koliko će se sabrati brojeva) se upisuje na početku. Zatim se postavljaju početne vrijednosti varijable s ($s:=0$) i varijable i ($i:=1$). **REPEAT** petlja počinje sa izrazom $s:=s+i$. Računar izvodi prvo desni dio ovog izraza. Vrijednost varijable s se uvećava za vrijednost varijable i ($s+i$). Ova vrijednost se pridružuje varijabli s . Zatim se uvećava vrijednost varijable i . Petlja se ponavlja sve dok vrijednost varijable i ne bude veća od n . Izračunata vrijednost s se ispisuje na kraju programa.

Tabela 4.10. Suma

Tekstualni algoritam	Listing programa:	Ekran
1. upis do broja (n) 2. početne vrijednosti sume s i kontrolne varijable i 3. formirati novu sumu ($s=s+i$) 4. uvećati kontrolnu varijablu i 5. kraj petlje 6. ispis sume	<pre>PROGRAM PP4_10; VAR i, n, s: Integer; BEGIN Write('Do broja'); Readln(n); s := 0; i := 1; REPEAT s := s + i; i := i + 1; UNTIL i > n; Writeln('Suma prirodnih brojeva do , n, ' je ', s); Readln; END.</pre>	Do broja 3 Suma prirodnih brojeva do 3 je 6

Daljnji rad: Izračunati proizvod prvih n prirodnih brojeva.

4.11. Naći proizvod prirodnih brojeva u intervalu od k do n . Zadatak riješiti pomoću **REPEAT** petlje.

Opis rješenja: Proizvod se izračunava u izrazu $p*i$. Početna vrijednost proizvoda je 1, to je jedinični element za operaciju množenja. Pri izvođenju programa računar množi vrijednosti varijabli p i i ($p*i=1*1=1$). Ova vrijednost se pridružuje varijabli p . Zatim se uvećava vrijednost varijable i za jedan ($i=k+1$). Ponovo se izvodi množenje varijabli i i p , te se izračunata vrijednost pridružuje varijabli p . Proces se ponavlja sve dok je i manje ili jednako n .

Opis programa: Prvo se upisuju granice intervala k i n , provjerava se da li je donja granica manja od gornje. Slijedi postavljanje početnih vrijednosti proizvoda p ($p:=1$) i kontrolne varijable i ($i:=k$). Prva naredba u **REPEAT** petlji je formiranje proizvoda ($p:=p*i$). Izračunata vrijednost se pridružuje proizvodu p . Slijedi uvećanje vrijednosti kontrolne varijable i ($i:=i+1$). Provjera ponavljanja se izvodi sa naredbom **UNTIL**. Ispis izračunatog proizvoda je na kraju.

Tabela 4.11. Proizvod prirodnih brojeva od k do n

Tekstualni algoritam	Listing programa:	Ekran
1. upis granice intervala (k, n) 2. ako su pogrešne granice ($k > n$), idi na 1 3. početna vrijednosti 4. formirati novi proizvod ($p=p*i$) 5. uvećati i ($i=i+1$) 5. ako je $i \leq n$ tada idi na 4 6. ispisati izračunati proizvod	<pre> PROGRAM PP4_11; VAR i, k, n, p: Integer; BEGIN REPEAT Write ('Granice '); Readln(k, n); UNTIL k < n; i := k; p := 1; REPEAT p := p * i; i := i + 1; UNTIL i > n; Writeln('Od ', k, ' do ', n, ' prizvod prirodnih brojeva je ', p); Readln; END. </pre>	<p>Granice 3 5</p> <p>Od 3 do 5 prizvod prirodnih brojeva je 60</p>

Daljnji rad: Izračunati sumu i proizvod u intervalu od K od N . Koristiti dvije petlje. (Zatim napisati drugu verziju sa jednom petljom).

4.12. Izračunati sumu parnih prirodnih brojeva u granicama k do n . Za generisanje prirodnih brojeva koristiti **REPEAT** petlju.

Opis rješenja: Suma se formira samo za parne vrijednosti prirodnih brojeva iz navedenog intervala. Parnost broja (djeljivost sa 2) se provjerava logičkim izrazom $i \bmod 2=0$. Logički izraz je istinit kad je broj i paran - nema ostatka pri dijeljenju sa 2, odnosno ostatak je nula. Za ove slučajeve (broj paran) formira se suma po izrazu ($s=s+i$) koji je objašnjen u navedenom zadatku.

Opis programa: Početna vrijednost sume s ($s:=0$) i varijable i ($i:=k$). Naredba **REPEAT** označava početak petlje. Prva naredba je provjera parnosti (djeljivost sa 2) kontrolne varijable i . Logički izraz $i \bmod 2=0$ je istinit kad nema ostatka pri dijeljenju tj. kad je broj paran. Za ove slučajeve se izvodi formiranje sume prema izrazu $s:=s+i$. Izračunata vrijednost izraza $s+i$ se pridružuje varijabli s . Zatim se uvećava vrijednost varijable i . Petlja se ponavlja sve dok vrijednost varijable i ne bude veća od n . Izračunata vrijednost varijable s se ispisuje na kraju programa.

Tabela 4.12. Suma parnih brojeva od k do n

Tekstualni algoritam	Listing programa:	Ekran
<p>1. upis granica (k, n)</p> <p>2. ako su pogrešne granice ($k>n$), idi na 1</p> <p>3. početne vrijednosti sume s i kontrolne varijable i</p> <p>4. ako je broj paran dodati ga na sumu</p> <p>5. uvećati kontrolnu varijablu i</p> <p>6. ako je $i>n$ tada idi na 4</p> <p>7. ispisati izračunatu sumu</p>	<pre>PROGRAM PP4_12; VAR i, k, n, s: Integer; BEGIN REPEAT Write ('Granice '); Readln(k, n); UNTIL k<n; s := 0; i := k; REPEAT IF i MOD 2=0 THEN s := s + i; i := i + 1; UNTIL i > n; Writeln('Suma parnih brojeva od ', k, ' do ', n, ' je ', s); Readln; END.</pre>	<p>Granice 5 11</p> <p>Suma parnih brojeva od 5 do 11 je 24</p>

Daljnji rad: Izračunati sumu neparnih i proizvod parnih brojeva u intervalu od k od n . Koristiti dvije **REPEAT** petlje.