

```
PROGRAM Maximumi;  
Uses wincrt;  
VAR a,b,max: INTEGER;  
BEGIN  
    IF a > b  
THEN  
    max:=a;  
ELSE  
    max:=b;  
WRITELN('Vlera maksimale eshte ',max);  
END.
```

P R O G R A M I M I

**PËR VITIN E DYTË PROFILI ELEKTROTEKNIK PËR
TEKNIKË KOMPJUTERIKE DHE AUTOMATIKË**

KUMANOVË, 2014

P Ë R M B A J T J A

H Y R J E	4
1. BAZAT E PROGRAMIMIT.....	5
1.1. SOFTUERI, TIPET E SOFTUERIT	6
1.2. ALGORITMET,PROGRAMIMI DHE GJUHËT PROGRAMUESE	9
1.2.1. ALGORITMET	9
1.2.2. GJUHËT PROGRAMUESE	12
1.2.3. GJUHËT E ULTA PROGRAMUESE	13
1.2.4. GJUHËT E LARTA PROGRAMUESE	15
1.3. TIPET E TË DHËNAVE	17
1.4. INTERPRETUESI DHE KOMPAJLUESI (PËRKTHYESI), MENITË NË TURBO PASCAL	18
1.4.1. INTERPRETUESI DHE KOMPJALUESI	18
1.4.2. MENITË NË TURBO PASCAL.....	19
PYETJE DHE DETYRA	21
2. TIPET THEMELORE TË TË DHËNAVE	23
2.1. ELEMENTET E GJUHËS PASCAL.....	24
2.2. STRUKTURA E NJË PASCAL PROGRAMI	25
2.3. NDRYSHORET, DEKLARIMI I TYRE.....	31
2.4. INICIALIZIMI DHE PARAQITJA NË EKRAN I NDRYSHOREVE	33
2.5. INICIALIZIMI I NDRYSHOREVE PËRMES TASTIERËS	35
2.6. DEKLARIMI I TIPEVE	37
2.7.1. KONSTANTET.....	38
2.7. TË DHËNAT LOGJIKE OPERATORËT LOGJIKE PARAQITJA E TYRE NË EKRAN	39
2.8. TË DHËNAT NUMËR I PLOTË, OPERATORËT.....	40
2.8.1. FUNKSIONET STANDARDE ME NUMRA TË PLOTW	46
2.9. NDRYSHORET E TIPIT NUMËR DHJETOR	50
2.9.1. FUNKSIONET STANDARDE ME NUMRA DHJETOR.....	54
2.10. PARAQITJA E NUMRAVE TË PLOTË NË KOMPJUTER	55
2.11. PARAQITJA E NUMRAVE DHJETOR ME PIKË LËVIZËSE	57
2.12. NDRYSHORET E TIPIT KARAKTER.....	59
2.13. TË DHËNAT E TIPIT VARG KARAKTERËSH (TEKST)	62

1.1.	OPERATORËT RELACIONAL	63
3.	CIKLET DHE DEGËZIMET NË PROGRAME.....	64
3.1.	STRUKTURAT PËR DEGËZIM IF	65
3.2.	STRUKTURA PËR DEGËZIM CASE.....	68
3.3.	STRUKTURAT CIKLIKE	70
3.3.1.	STRUKTURA WHILE-DO	70
3.3.2.	STRUKTURA REPEAT-UNTIL.....	70
3.3.3.	STRUKTURA FOR- TO-DO	72
3.3.4.	STRUKTURA FOR-DOWNTWO-DO	72
	Referencat.....	74

HYRJE

Programimi për vitin e dytë është bërë në bazët të planit mësimor nga qendra për arsim profesional për lëndën e programit të vitit të dytë për shkollat e mesme profesionale drejtimi Elektroteknik për teknikë kompjuterike dhe automatikë. Qëllimi i tekstit është që të mësohen bazat dhe principet themelore të PROGRAMIMIT.

Materiali mësimor sqaron konceptet elementare dhe është i përshtatshëm për të gjithë fillestarët që dëshirojnë të njihen me bazat e programimit, dhe gjuhën programuese Pascal. Versioni i definuar në planprogramin mësimor është Turbo Pascal 1.5, mirëpo në numër të konsiderueshëm të rasteve mund të përdoren edhe versione tjera të gjuhës programuese Pascal.

Për të filluar mësimet në këtë tekst mësimor nevojiten njohuri elementare nga limitët: algoritmet, aritmetika, operacionet themelore aritmetikore, numrat binar dhe operacionet matematikore me numra binar, njohja e punës me sistem operativ Windows.

Materiali është ndarë në tema dhe njësi më të vogla mësimore me qëllim që nxënësi dhe mësimdhënësit të i lehtësohet puna dhe arrihen suksese në mënyrë më të shpejtë dhe më të lehtë.

Libri për çdo detal të ri që sqarohet për të qenë më i afërt me nxënësin dhe për ti patur më të qartë do të jipet edhe shembulli konkret i implementuar dhe sqaruar detalisht, në mënyrë që të jetë sa më e qartë dhe konkrete për fillestarët e programimit.

Në kapitullin e parë sqarohen elementet kryesore të softuerit dhe tipet e gjuhëve programuese. Sqarohet sistemi kompjuterik pjesët themelore të një sistemi si hardueri dhe softueri. Definohen nocionet harduer dhe softuer duke i përshkruar me shembuj, dhe duke e qartësuar relacionin e harduerit, softuerit dhe programimit të një sistemi kompjuterik. Për të përvetësuar temën nevojiten njohuri nga lëmia e informatikës, kuptimi i nocioneve, termeve dhe funksioneve themelore për sistemin kompjuterik.

Nderim Rahmani

1.BAZAT E PROGRAMIMIT

Kapitulli 1

Qëllimet e temës

- Të njohë ndarjen e pjesës softuerike të njehsorit (gjuhët programuese, programet aplikative dhe shërbyese);
- Të njohë ndarjen e gjuhëve programuese (gjuha e makinës , assembler dhe gjuhët e larta programuese);
- Të kuptojë programimin;
- Të njohë ndarjen e tipeve të të dhënave në programim;
- Të shpjegojë kompajler dhe interpreter;
- Të dallojë kompajler dhe interpreter;
- Të punojë me kompajler TPË 1.5 (TURBO PASCAL FOR WINDOWS)
 - (Menite: Ndhmë fajlat nëpascal, kompajlim-debagim).

Njohuritë e nevojshme për përvetësimin e temës

Njohuri bazë në lëminë e informatikës: harduerit dhe softuerit

1.1. SOFTUERI, TIPET E SOFTUERIT

Sistemet kompjuterik paraqesin pajisje të cila në ditët e sotme hasen në çdo pjesë të përditshmërisë duke përfshirë nevojën e përdorimit të tyre në shumë profesione (inzhinieri, mjeksi, ekonomi, arsimi, etj), për zbavitje, komunikim, automobilitë, aparate në shtëpi dhe shumë lloje të pajisjeve. Vetia e përbashkët e sistemeve kompjuterike është se ato përbëhen nga dy pjesë themelore: hardueri dhe softueri të cilët punojnë në harmoni me njëri tjetrin për të kryer funksione të dobishme për shfrytëzuesin e sistemit kompjuterik.

Softueri dhe hardueri paraqesin dy komponente themelore dhe në kuptim funksional të pandara të çdo sistemi kompjuterik.

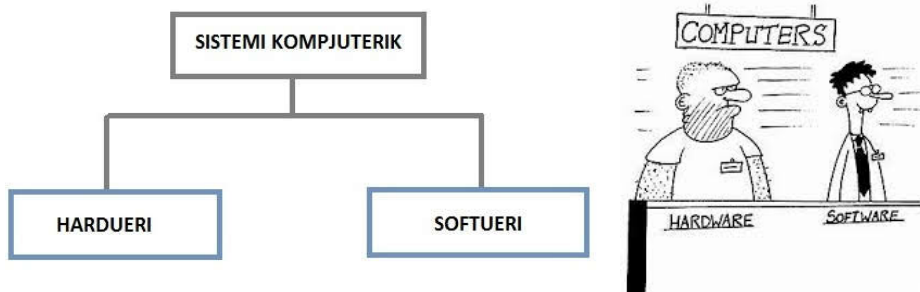


Figure 1. Sistemi kompjuterik

Hardueri (nga anglishtja hard- e fortë dhe ware-produkt) paraqet pjesën mekanike, elektronike të kompjuterit dhe pajisjet për lidhjen në mes tyre, ku përfshinë pllakën amë, pajisjet e vendosura në pllakën amë, pajisjet hyrese dhe dalëse tastiera, miu, altoparlantët dhe gjitha pjesët tjera mekanike që mund ti shohim me sy ose ti prekim me dorë.



Figura 2 Hardueri

Softueri (nga anglishtja soft - e butë dhe ware-produkt) përfshin bashkësinë e programeve kompjuterike dhe të dhënat e tjera që përmbajnë instruksione që i kontrollojnë punën e pjesëve harduerike në përpunimin, leximin dhe ruajtjen e informacioneve. Gjithashtu mund të themi se softueri i referohet një apo më shumë programeve kompjuterike, lojrave dhe të gjitha të dhënave të ruajtura në memorien e kompjuterit.

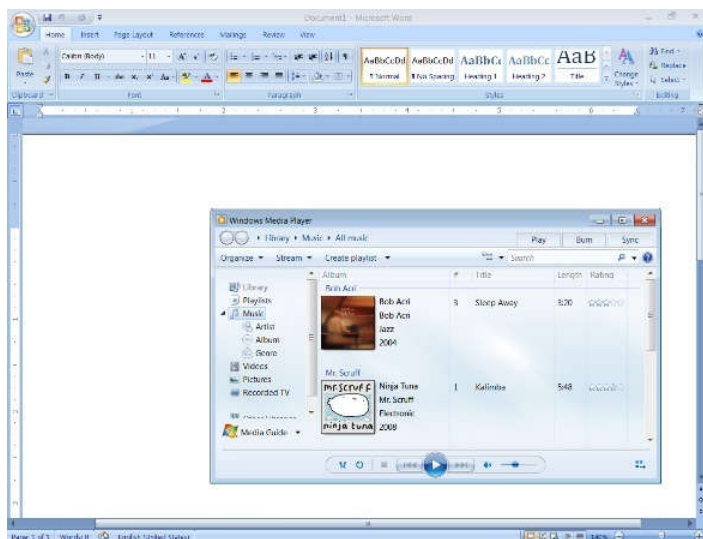


Figura 3 Softueri

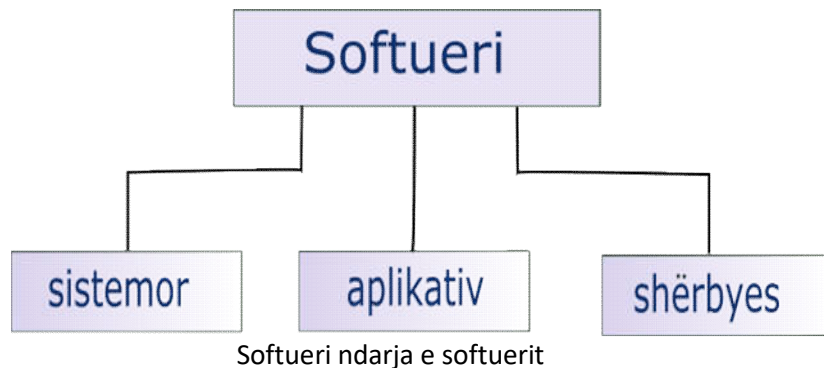
Softueri kryen funksionin e programit ose duke i dhënë instruksione të drejtëpërdrejta harduerit të kompjuterit ose duke shërbyer si lidhje në mes të programeve të tjera dhe harduerit.

Shembuj të softuerit mund të jetë Ëindoës, Linux, Microsoft Word, Microsoft Excel, Microsoft Power Point, Avira Antivirus, Photoshop, C++, Google Chrome, Winamp, Media player. Anlogjikisht hardueri pa softuer mund të thuhet se është libër i pashkruar, teksti i shkruar në libër e bën librën të përdorshëm.

Nga shembujt e përmendur shohim se softueri përfshin një spektër të gjërë të përorimit në shumë fusha. Gjithashtu dokumentet tona të shkruara në MS Word, fotografia jonë e ruajtur në kompjuter, kengët tona të preferuara paraqesin pjesë të softuerit. Nga mund të definojmë se:

Softueri paraqet bashkësinë e instruksioneve dhe të dhënave që ruhen në sistemin kompjuterik,

Softuerin ndahet nw tri kategori: *sistemor, aplikativ dhe shërbyes*



SOFTUERI SISTEMOR përmbanë: sistemin operativ dhe të gjitha veglat të cilat e mundësojnë funksionimin e sistemit kompjuterik. Menaxhimin me të gjitha resurset harduerike, menaxhim me të dhënat dhe egzekutimin e aplikacione të shfrytëzuesit.

Shembuj të sistemit operativ janë, Windows Vista, Windows 7, Windows 8, Linux, Macintosh OS X, Android, etj. Sistemi operativ është softueri i parë e cila instalohet në një sistem kompjuterik, pa të cilin sistemi kompjuterik do të ishte i pa përdorshëm.

SOFTUERI APLIKATIV i cili përmbanë programet të cilat kryejn punët e shfrytëzuesve, psh. tekst procesorët (word processors), tabelat (spreadsheets) dhe sistemet për menaxhim me bazat e të dhënave (databasemanagement systems), bien në kategorinë e softuerit aplikativ.

Softueri aplikativ përfshin atë klasë të programeve me të cilat ndërmerret dhe njerëzit me ndihmën e kompjuterëve zgjedhin probleme/detyra konkrete nga sfera e interesimit të tyre. Programet të cilat realizojnë porositë, kontrollojnë tabelat, përpunojnë të dhëna, lundrojnë në internet, dëgjojnë muzikë, përpunojnë tekst, sigurojnë informata për marketing e financa etj. paraqesin vetëm disa shembuj të softuerit aplikativ që ofrohet në tregun softuerik.

Shembuj të softuerit aplikativ janë Paketa Office që përfshin Microsoft Office Word, Microsoft Office Excel, Microsoft Office Power Point, Microsoft Office Access, Winamp, Mozilla Firefox, Internet Explorer, Skype, Photoshop etj.

SOFTUERI SHËRBYES përbëhet nga programet të cilat shfrytëzohen për vendosjen e parametrave punuese të kompjuterit, për testimin e tij, duke bërë kopje rezervë prej dokumenteve, kompresim të të dhënave, programe antivirus etj,. Me qëllim që të "shërbej" shfrytëzuesit dhe lehtësoj kryerjen e detyrave të përmendura ose të ngjashme. Janë të

njohura edhe si vegla programore. Të tilla janë Avira Antivirus, WinZip, Backup, Disk Defragment dhe shumë aplikacione tjera.

Shkurtimisht:

- Çdo sistem kompjuterik përbëhet nga dy pjesë themelore hardueri (pjesa e forte dhe e prekshme nga dora e njeriut) dhe softueri (aplikacionet dhe të dhënat)
 - Softueri ndahet në dy pjesë: softuer sistemor dhe aplikativ
 - Softueri sistemor bën menaxhimin me të gjitha resurset harduerike, aplikacionet dhe të dhënat në një sistem kompjuterik
 - Softueri aplikativ përmban një spektër të gjërë të aplikacioneve të cilat përdoren për të kryer detyra të ndryshme për shfrytëzuesin e sistemit kompjuterik.
-
- Në tekst janë të cekura vetitë dhe funksionet themelore të sistemeve operative, të gjenden funksionet dhe vetite specifike të tyre
 - Të gjenden aplikacione të cilat paraqesin softuer aplikativ dhe të përshkruhen detyrat e tyre, lëmia në të cilën gjen zbatim.

1.2. ALGORITMET, PROGRAMIMI DHE GJUHËT PROGRAMUESE

1.2.1. ALGORITMET

Softueri ose programet aplikative kompjuterike përdoren për të zgjidhur probleme ose për të shpejtuar zgjidhjen e problemit. Por për të zgjidhur problemin është e rëndësishme që të ndërmerren hapa të caktuar gjegjësisht detyra. Por çdo problem për tu zgjidhur duhet të dihen hapat ose detyrat që duhet të egzekutohen mbi të dhënat numerike, tekstuale, grafike prej problemeve elementare deri te problemet që përbëhen nga llogaritje komplekse. Që nga detyra e mbledhjes ose zbritjes së dy numrave, operacionet me thyesa, shuma ose ndryshimi i dy kohëve të dhëna me orë dhe minuta, gjetja e shumes së vargut me më shumë numra; duhet që

kompjuterit ti "sqarojmë" detalisht çdo hap për ta zgjidhur çdonjërin nga detyrat (problemet e parashtruara). Softueri i cili zgjidh këto probleme duhet të shkruhet në makinën llogaritëse në atë mënyrë që të zgjidhja e problemeve të jetë e saktë. Por zakonisht para se të shkruajmë instruksionet e zgjidhjes së detyrës në kompjuter zakonisht duhet të mendojmë dhe analizojmë **hapat** e zgjidhjes së detyrës.

Shembull: Të shkruhen hapat të cilët mundësojnë gjetjen e dallimit në mes të dy kohëve të dhënë në orë dhe minuta Koha e parë është më e vonshme se koha e dytë?

Zgjidhje:

Kohët e dhëna do ti emërojmë si **Koha1** dhe **Koha2** të cilat janë dhënë në orë dhe minuta **H1:M1** dhe **H2:M2** përkatësisht, dhe rezultate të tregohen në kohën e emërtuar si **Koha** në formatin **H:M** (mendon se si do ta kishit caktuar në mënyrë të pavarur).

Atëherë:

Koha1: 05:45

Koha2: 04:35

Dallimi i kohëve llogaritet si:

05 : 45

- 04 : 35

01 : 05

Dallimi mes kohëve Koha 1 dhe Koha 2 është 1 orë dhe 5 minuta

Por nëse kohët e dhëna kanë vlerat:

Koha1: 06:20

Koha2: 04:45

Atëhere nga shprehja;

06 : 20

-04 : 45

02 :-25

Shohim se minutat e kohës së parë janë më pak se minutat e kohës së dytë dhe dallimi i tyre është negativ. Në këtë rast minutave të kohës së parë duhet ti shtohen për 60 minuta të cilat merren nga ora, ndërsa ora e kohës së parë të zvogëlohet për 1 orë Koha e parë merr formën si 05:80 dhe operacioni kryhet si:

05 : 80 (80=20+60)

-04 : 45

01 : 35

Nga shohim se dallimi mes kohëve të dhëna është 01 orë dhe 35 minuta.

Detyra jonë është që të përshkruajmë hapat që duhet të ndërmmiren në rastin e përgjithshëm për përcaktimin e dallimit të dy kohëve të dhëna në orë dhe minuta. Prandaj zgjidhjen do ta përshkruajmë detalisht.

Para së gjithash do të kemi emërtimin e vlerave me shkurtesa për minutat dhe orët me:

*M1 –minutat për kohën e parë,
H1- ora për kohën e parë,
M2- minutat për kohën e dytë,
H2- ora për kohën e dytë,
H-dallimi i kohëve në orë
M- dallimi i kohëve në minuta
atëherë hapat që duhet të ndërmerren do të ishin:*

1. Nëse $M1 > M2$ atëherë kalo në hapin 3
2. Vlera $M1$ zmadhohet për 60 minuta dhe vlera $H1$ zvogëlohet për 1 orë e shprehur matematikisht $M1=M1+60$ dhe $H1=H1-1$
3. Cakto ndryshimin e $M1$ dhe $M2$ dhe ruaje në ndryshoren M ; $M1=M1-M2$
4. Cakto ndryshimin e orëve $H1$ dhe $H2$ dhe ruaje në ndryshoren H ; $H=H1-H2$
5. Rezultatin shfaq si $H:M$

Nga kjo mund të shihet se për të zgjidhur një problem sado i vogël në rastin tonë caktimi i dallimit të dy kohëve duhet të kemi një listë me të dhëna ($H, M, H1, M1, H2$ dhe $M2$) dhe një listë me instruksione (hapi 1 deri te hapi 5) të cilat janë të radhitura fiks. Çfarë do të ndodhë nëse radhitja do të ishte ndryshe? Për shembull nëse hapi i parë dhe hapi i katërt do të ndërrojnë vendet me njëri tjetrin. Rrjedhimisht në shumë raste rezultati nuk do të ishte i saktë, që do të thotë se radhitja e urdhërave është domosdoshmëri që rezultati i detyrës të caktohet saktësisht.

Veprimet e shkruara fiks si në shembullin e mësipërm njihen si algoritme, të cilat janë edhe bazë në shkruarjen e programeve me kompjuter, dhe në definicioni i tij është:

Algoritmi paraqet grumbull të veprimeve me një radhë të fiksuar, të cilët ndërmerren gjatë zgjidhjes së një problemi të caktuar.

Shembuj tjerë të algoritmeve mund të jenë receta për gatimin e një ushqimi, dërgimi i një porsie përmes telefonit dhe shumë probleme tjera që hasen në përditshmëri.

E rëndësishme për algoritmet është të thuhet se ato mund të jenë **detal** ose të **përgjithshëm**, varësisht nga ajo se hapat e zgjidhjes jipen hollësisht apo jo.

Në rastin e programimit kompjuteri për të zgjidhur një problem i nevojitet “lista e përbërësve” ose të dhënat, të cilat mund të jenë të shprehura në disa forma si: numra, tekste, zë, fotografi të cilat janë të ruajtura në kompjuter që më vonë këta “përbërës” do ti quajmë ndryshore. Ndryshoret mund të përfaqësojnë të dhëna të cilat përpunohen dhe ruhen në kompjuter në forma të ndryshme si të dhëna numerike, tekst, zë ose fotografi.

Përpunimi i të dhënave në kompjuter bëhet përmes programeve të veçanta të cilët shkruhen nga programuesit. Çdo program mund të thuhet se është algoritëm i shëndruar në urdhëra.

Shembuj të programeve ose softuerit kompjuterik kemi të cekur paraprakisht ku mund të vërejmë se detyra e softuerit është përpunimi, ruajtja, bartja i të dhënave të ndryshme në bazë të rregullave të caktuara paraprakisht ose të dhëna nga shfrytëzuesi.

Shkurtimisht mund të definohet si:

Një program kompjuterik, ose program, paraqet listë të instruksioneve e shkruar për ta “kuptuar” dhe egzekutuar kompjuteri që të kryej detyrë të caktuara.

Programimi paraqet procesin zhvillimit dhe implementimit të bashkësive të ndryshme të instruksioneve që kompjuteri të kryej detyrë të caktuara. Ose shkurtimisht programimi paraqet procesin e shkruarjes së programit.

1.2.2. GJUHËT PROGRAMUESE

Programi kompjuterik është bashkësi e urdhërave të cilat egzekutohen në kompjuter. Urdhërat të cilët i jipen kompjuterit shkruhen duke u bazuar rregullave të caktuara të cilat kompjuteri për ti egzekutuar duhet që ti “kuptojë”. Bashkësia e rregullave të cilat mundësojnë përpilimin e urdhërave të kuptueshëm për sistemin kompjuterik paraqet gjuhën programuese.

Gjuha programuese është gjuhë e cila përmban një bashkësi të rregullave me anë të cilave i urdhërojmë kompjuterët të kryejnë detyra të caktuara. Qëllimi i gjuhës programuese është që makinës llogaritëse (kompjuterit) ti jepen instruksionet të shkruara të cilat do të egzekutohen për të kontrolluar sjelljen e kompjuterit ose të shprehin algoritme.

Një gjuhë programuese mundëson që të përpilohen programe për kompjuter, të cilat mundësojnë që kompjuteri të kryej operacione llogaritëse, kontrollojë hyrjen dhe daljen e të dhënave, kontrollojë pajisjet e kompjuterit si printer, hard disk, dhe shumë funksione tjera të dobishme për shfrytëzuesit e sistemit kompjuterik.

Egzistojnë mijëra gjuhë të ndryshme programuese të krijuara të cilat mundësojnë krijimin e aplikacioneve të shumëllojshme.

Të gjitha shembujt e cekur të softuerit sistemor dhe aplikativ janë të krijuara me anë të gjuhëve programuese.

Të dhënat të cilat përpunohen dhe ruhen në kompjuter ruhen me shifrat "0" dhe "1", do të thotë se kompjuteri gjatë përpunimit të të dhënave njih vetëm dy simbole. Gjithashtu instruksionet në nivelin më të ulët jipen vetëm me shifrat "0" dhe "1". Kombinimi i tyre mundëson përpilimin e urdhërave dhe egzekutimin e operacioneve komplekse me të dhëna të cilat konvertohen në formë të kuptueshme për shfytëzuesin e sistemit kompjuterik.

Varësisht nga ajo se urdhërat e shkruara janë një gjuhë e afërt me parimin e punës së kompjuterit apo të njeriut dallojmë këto lloje të gjuhëve programuese.

- Gjuhë të ulta programuese dhe
- Gjuhë të larta programuese

1.2.3. GJUHËT E ULTA PROGRAMUESE

Karakteristikë të gjuhët e ulëta programuese është se instruksionet në gjuhët e ulëta programues jipen drejtëpërdrejtë në processor, kanë qasje të drejtëpërdrejtë tek regjistrat e procesorit dhe memorja, programet e shkruara në këtë gjuhë posedojnë performanca të shkëlqyera gjatë egzekutimit.

Gjuha makinës është përfaqësues tipik dhe kryesor i gjuhëve të ulëta programuese. Urdhërat në gjuhën e makinës përbëhen prej bitave (binary digits), 1 dhe 0 dhe të cilën në fakt e "kupton" procesori i kompjuterit dhe me anë të cilës në fillimin e zhvillimit të kompjuterëve janë shkruar programet e para. Çdo urdhër i programit është shkruar vetëm me ndihmën e shifrave binare.

Shembull: Të llogaritet shuma e numrave **a** dhe **b** dhe të ruhet në **c**.

Kompjuteri duhet të njehsojë shumën e dy numrave të ruajtur në memorjen e tij të "emërtuara" si **a** dhe **b** dhe të llogarisë shumën e tyre në memorjen e rezervuar **c**. Kompjuteri për çdo adresim ose urdhër përdorë numrat binar.

Nëse adresa e rezervuar për **a** është 111001, për **b** është 101011, për **c** është 100011 operatori për mbledhje (+) duhet të shkruhet si 111101 urdhëri për lexim të ndryshores është 111000, urdhëri për ruajtjen e të dhënës në një adresë është 000111 atëherë, pjesë nga programi mund të shkruhej si:

111000111001

```
000111 100011
111000101011
111101 101011100011;
```

“Programi” i dhënë i cili cakton shumën e dy numrave dhe e rezultatin e fituar e ruan në një lokacin të memorjes është duket i pakuptueshëm. Gjithashtu, pasi që gjuha e makinës është e vështirë për t’u kuptuar dhe zbatuar, dhe është e vështirë që komandat (udhezimet) të mbahen mend (çdo udhëzim duhet të shënohet me anë të numrave 1 dhe 0), janë krijuar gjuhët programuese të cilat në fakt janë përkthim i urdherave me bita (1 dhe 0) që mundet të mbahet në mend lehtë dhe me anë të së cilës mund të programohet.

Kjo mënyrë e programimit ka qenë shumë e vështirë prandaj inxhinierët kanë përsosur mënyrën e programimi duke krijuar gjuhën assembler.

Gjuha assembler i takon gjuhëve të nivelit të ulët të gjuhëve programuese për një kompjuter ose pajisje të tjera programueshëm, në të cilën ka një korrespondence (lidhje) shumë të fortë (përgjithësisht një-për-një) midis gjuhës dhe arkitekturës së kodit të makinës.

Kodi i gjuhës assembler konvertohet në kod të ekzekutues të makinës nga programi i njohur si assembler; Urdhërat e shkruara në gjuhën assembler janë më të afërta me gjuhën e njeriut, dhe mundëson që programuesi për të shkruar urdhëra dhe operacione përdorë shprehje nga gjuha angleze, gjithashtu mundëson përdorimin e ndryshoreve.

Shembull:Të llogaritet shuma e dy vlerave **a** dhe **b**dhe rezultati të ruhet në memorjen e rezervuar **c**.

```
mov ax,a      ;vendose vlerwn e ruajtur num1 në ax
add ax,b      ;cakto shumwn e dy numrave
mov c,ax      ;vendose shumën e dy numrave në ndryshoren c
```

Gjuha assembler urdhërrat i egzekuton sipas radhës ashtu edhe si janë shkruar ku urdhëri i parë:

mov ax, a mundëson që vlerën e ruajtur në lokacionin a të vendoset në adresën e memorjes e quajtur ax

urdhërin e dytë add ax,b mundëson që vlerës ax i shtohet vlera e ruajtur në b.

Me urdhërin e tretë mov c,ax mundësohet që në në memorjen c të ruhet vlera e ax.

Me tre urdhërat arrihet që të caktohet shuma e numrave të ruajtur në a dhe b dhe shuma e tyre të ruhet në memorjen c.

Dallimi mes gjuhës së makinës dhe gjuhës assembler është i qartë, gjuha assembler është më e afërt me gjuhën natyrore (gjuhën angleze) , më e lehtë për ta kuptuar për njeriun. Më këtë edhe më lehtë shkruhen programet dhe më lehtë kuptohen programet e shkruara më parë.

1.2.4. GJUHËT E LARTA PROGRAMUESE

Gjuhët programuese mund të përdoren për të krijuar programe që kontrollojnë sjelljen e një dhe makinë apo për të shprehur algoritme në mënyrë precize.

Urdhërat në gjuhët e larta programuese nuk përshkruajnë detalisht instruksinet që duhet të egzekutojë kompjuteri sikurse në rastin e gjuhëve të ulëta programuese por jipen në nivel më të lartë, në të njejtën kohë duke qenë më i afërt me gjuhën e njeriut.

Gjuhët e larta programuese janë më të lehta për të programuar dhe kuptuar programi i shkruar pasi që përdorin shprehja nga gjuha angleze, sikurse shprehjet if, for, Begin, end; operacionet aritmetikore u janë përshtatur me ato që përdorim në praktike, +,-,*,<,>, DIV,MOD, dhe shprehje tjera të afërta me gjuhët natyrore të njeriut.

Shembull: Të caktohet shuma e vlerave **a** dhe **b** dhe rezultati të ruhet në ndrshoren c.

Urdhëri që do ti jipet gjuhës së lartë programuese do të duket si në vijim

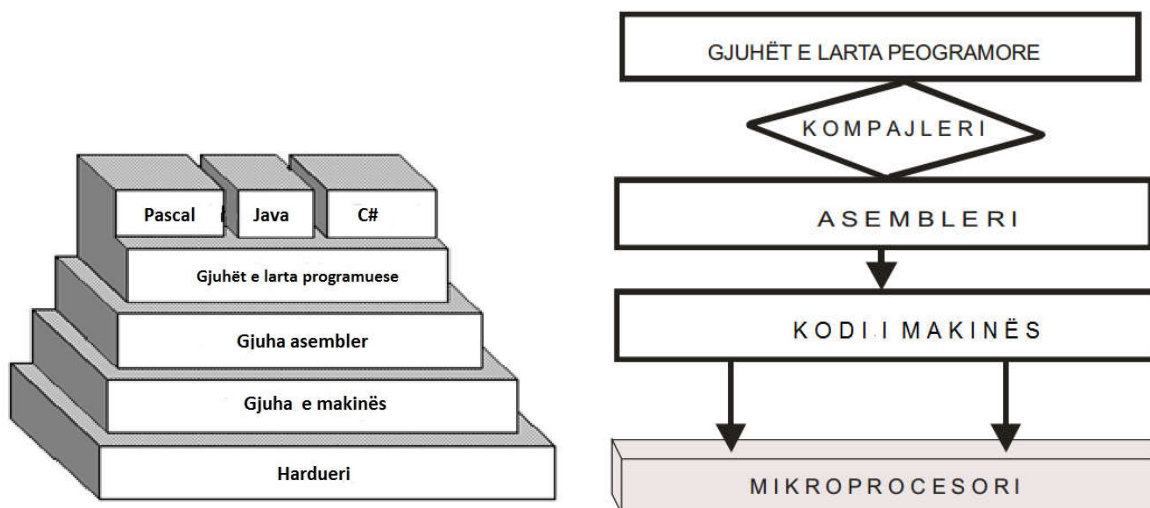
```
c=a+b;
```

Dallimi mes gjuhës assembler dhe gjuhës së lartë programuese shembujt e mësipërm e bëjnë të qartë. Edhe pse rezultati është i njejtë në të dy rastet, urdhërat në gjuhën e lartë programuese janë më të afërta me gjuhën natyrore të njeriut dhe më të lehta për tu shkruar dhe kuptuar nga programuesit.

Sot, janë një numër i konsiderueshëm i gjuhëve programore. Ndër gjuhët më të njohura janë: C++, Java, Pascal, Delphi, C#, php etj...

Përparësitë e gjuhëve të larta programuese janë:

- Urdhërat lehtë shkruhen, kuptohen
- Egzekutimi nuk varet nga lloji i procesorit
- Përfshijnë domen të gjërë të të dhënave



Hierarkia e gjuhëve programuese

Në disa tekste mund të haset se gjuhët programuese ndahen në tri kategori (gjuhë e ulëta , gjuhët e mesme dhe gjuhët e larta programuese)

Shkurtimisht

- *Algoritmi paraqet grumbull të veprimeve me një radhë të fiksuar, të cilët ndërmerren gjatë zgjidhjes së një problemi të caktuar. Algoritmi mund të jetë detal ose i përgjithshëm*
- *Programi kompjuterik paraqet listë të instruksioneve e shkruar për ta “kuptuar” dhe egzekutuar kompjuteri që të kryej detyrë të caktuar.*
- *Programimi paraqet procesin zhvillimit dhe implementimit të bashkësive të ndryshme të instruksioneve që kompjuteri të kryej detyrë të caktuar.*
- *Gjuhët e programimit përdoret për të krijuar programe që kontrollojnë sjelljen e një makine (kompjuteri) dhe/ose të shprehin algoritme të përshkruar saktësisht.*
- *Gjuhët programuese ndahen në gjuhë të larta dhe gjuhë të ulëta programuese*
- *Gjuhët e ulëta programuese janë të përafërta me gjuhën e kompjuterit dhe janë më pak të kuptueshme për njeriun, shprehja e algoritmeve me këto gjuhë është më e vështirë*
- *Gjuhët e larta programuese janë më të afërta me gjuhët natyrore, më të kuptueshme për njeriun , shprehja e algoritmeve është më e lehtë.*

Duke përdorur TIK të gjenden shembuj për Algoritme të ndryshme, mënyrat e paraqitjes së tyre.

1.3. TIPET E TË DHËNAVE

Pasiqë kompjuterët karakterizohen me përpunimin e të dhënave si objekti kryesor në programim janë vetë të dhënat. Në aplikacione të ndryshme hasim të dhëna të tipit të ndryshme. Nëse një program aplikativ do të duhej të përpunonte të dhënat e entitetit nxënës i shkollës do të kishim të dhënat duke filluar nga numri i ditarit, emri, mbiemri, data e lindjes, gjinia, notat e tij, mesatarja e notave të tij, ku vërejmë se do të operojmë me disa tipe të të dhënave si numra të plotë, numra dhjetor, listë të numrave, shkronja, tekst, datë.

Në praktikë hasim tipe të ndryshme të të dhënave si numra të plotë, real, të dhëna të tipit logjik, shenjë. Integer, real, dhe njihen si të dhëna tip I të dhënave standard.

Të dhënat mund të ndahen edhe nga struktura dhe tipi i tyre dhe varësisht se si ruhen nga kompjuteri. Në gjuhën programuese Pascal tipet e të dhënave mund ti përmbledhim me tabelën në vijim:

TË DHËNAT				
STATIKE			DINAMIKE	
Linearisht të pastrukturuar		Të strukturuar	Me madhësi të ndryshueshme	Me strukturë të ndryshueshme
Linearisht të radhitura	Linearisht të paradhitur	Vargu me bit		Lista
Kardinale	Real	Vargu me shenja		Pema
Numrat e plotë	Kompleks	Vektor i përgjithshëm		Tregues
Shenjë		Bashkësi		Object
Logjik		Record	Stek	
Të numërueshem			Vargu	
Interval			Fajlli	

Tipi statik i të dhënave janë me madhësi të përcaktuar më parë, vendosen në lokacione fikse të memorjes dhe për atë qasja deri tek ato është shumë e shpejtë.

Tipi dinamik i të dhënave janë ato të dhëna madhësia ose struktura e të cilave ndryshon gjatë ekzekutimit të programit. Ato vendosen në lokacione cila në momentin e krijimit të tyre është e lirë.

Tipet statike të pastrukturuara janë të dhëna elementare të cilat nuk mund të zbërthehen në të dhëna më të thjeshta kuptimplota

Linearisht të radhitur janë ato tipe vlerat e të cilëve mund të pasqyrohen në ndonjë interval të numrave të plotë.

Linearisht të paradhitur nuk e kanë vetnë e pasqyrimin në ndonjë interval të numrave të plotë.

Tipet e strukturuar të të dhënave janë tipe të përbëra që përmbajnë tipe elementare të të dhënave dhe/ose tipe të të dhënave të pastrukturuar.

Tipe dinamike të të dhënave mund të jenë me madhësi të ndryshueshme dhe/ose strukturë të ndryshueshme

Disa nga tipet e përmendura të të dhënave do ti përdorim në vazhdim të tekstit mësimor.

1.4. INTERPRETUESI DHE KOMPAJLUESI (PËRKTHYESI), MENITË NË TURBO PASCAL

1.4.1. INTERPRETUESI DHE KOMPJALUESI

Programi i shkruar në gjuhën e lartë programuese Pascal për tu egzekutuar duhet të analizohet nëse ka gabime dhe të “përkthehet” në gjuhën e makinës me qëllim që instruksionet e dëshiruara të egzekutohen nga procesori dhe krijohet programi i cili do të mund të egzekutohet në kompjuterë të ndryshëm nga shfrytëzuesi.

Të sqarojmë disa koncepte mbi egzekutimin e programit të shkruar në një gjuhë të lartë programuese. Siç e dimë e vetmja gjuhë të cilën e kupton kompjuteri është është e ashtuquajtura gjuhë e makinës. Gjuhë e larta programuese janë gjuhë përshtatura gjuhëve natyrore të cilat i kupton njeriu. Për këtë arsye nevojitet një gjuhë (aplikacion) e ndërmjetme e cila i transformon instruksionet e shkruara në gjuhën programuese në gjuhë të makinës. Egzistojnë dy lloje të këtyre gjuhëve që dallojnë për nga principi i punës.

Interpretuesi është përkthyes i cili çdo rresht nga kodi burimor e përkthen në gjuhën e makinës dhe e egzekuton. Nëse një rresht programor përsëritet më tepër here, e njëjta do të analizohet atëherë sa të paraqitet dhe do të përkthehet në gjuhën e makinës gjithashtu atëherë me qëllim që të egzekutohet. Kjo është arsyeja për të cilën shfrytëzimi i interpretuesit e ngadalëson egzekutimin e programit. Nga ana tjetër interpretuesi karakterizohet nga thjeshtësia për, për

shkak se është e mundur në moment të zbulohen gabimet eventuale, gjatë egzekutimit të programit. Qëllimi i përdorimit të interpretuesit nga programuesi është gjetja e gabimeve në kodin e programit, pasi që interpretuesi mund të paraqitjet rezultate të ndërmjetme gjatë egzekutimit të programit.

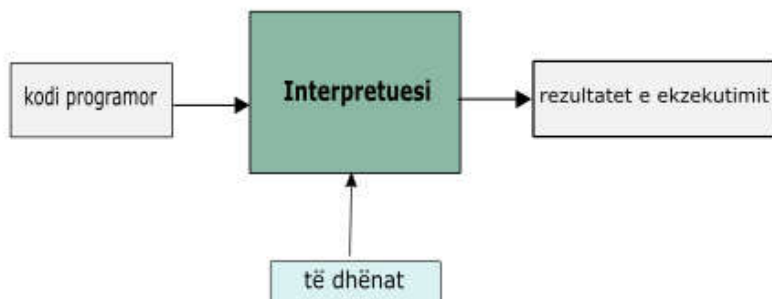


Figure 4 Interpretuesi

Kompajluesi (përkthyesi ose përpiluesi) është program i cili çdo urdhër të kodit burimor e analizon dhe e përkthen në kodin e makinës, e regjistron në fajll dalës egzekutues. Gjatë egzekutimit egzekutohen instruksionet e përkthyer në kodin e makinës gjegjësisht fajllin egzistues. Kjo është arsyeja që programet egzekutohen më shpejtë për dallim nga kur egzekutimi i programit bëhet përmes interpretuesit. Nëse instruksionet e shkruara përmbajnë gabime sintaksore atëherë kompajluesi nuk mund të krijojë programin egzekutiv.

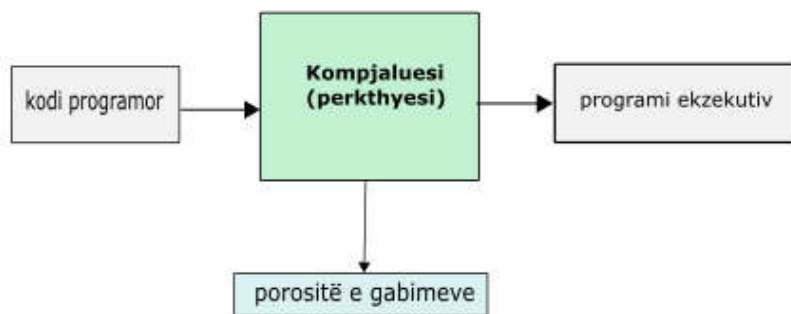


Fig x. Kompajluesi

1.4.2. MENITË NË TURBO PASCAL

Turbo Pascal është gjuhë programuese e cila na mundëson krijimin e programeve. Pas startimit të programit Turbo Pascal në ekran do të paraqitet dritarja aplikacionit.

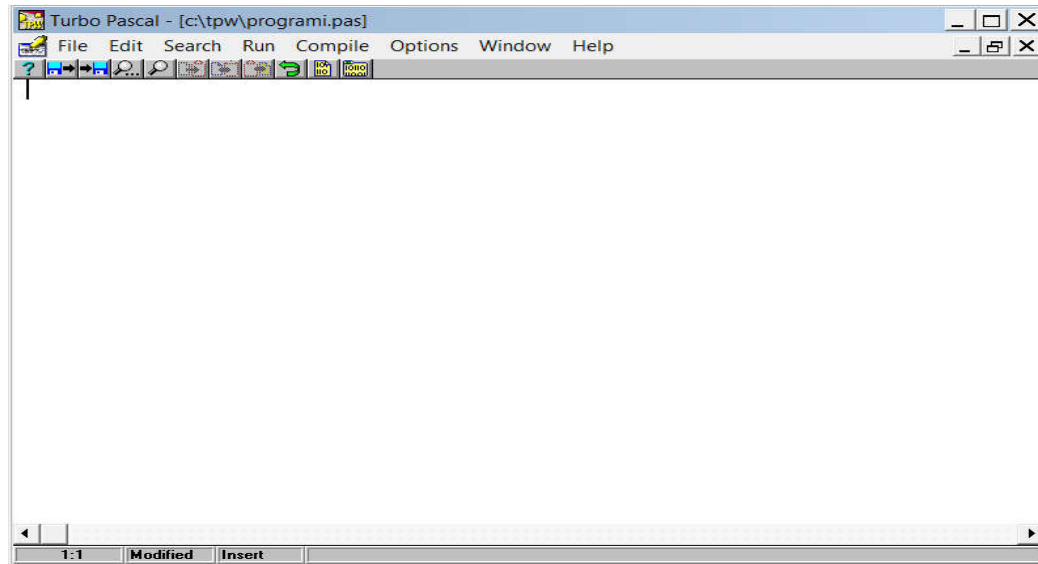


Figure 5 Dritarja e aplikacionit të Turbo Pascalit

Turbo Pascal ka një rrethinë zhvilluese e cila mundëson programuesit të krijojë, redaktojë, kompajlojë, lidh, egzekutojë dhe të **gjejë gabimet**.

Mbyllja e dritares së Turbo Pascalit bëhet me **X** në anën e djathtë të dritares.

Në listën në vijim do të sqarojmë shkurtimisht disa nga opsionet elementare për zhvillimin e programeve në Turbo Pascal janë:

Opsionin File

- **New**- mundëson krijimin e fajlit të ri
- **Open** – hapja e fajlit egzistues
- **Save** – ruajtja e ndryshimeve ose ruajtja e fajlit (fajlat në gjuhën programuese Pascal ruhen me prapashtesën .pas)
- **Save as**- ruajtja e fajlit me një emër të ri
- **Exit**- mbyllja e programit

Opsionin Run

- **Run**- ekzekuton programin
- **Trace into**- fillon egzekutimin e programit vijë për vijë, ose vazhdon egzekutimin në urdhërin e ardhëshëm.
- **Step over** - egzekuton programin urdhër pas urdhëri duke marrë nënprogramin si një urdhër të vetëm.

Opsioni Compile

- **Compile**- gjen dhe tregon gabimet sintaksore në kodin programor , dhe nëse është pa gabime krijon fajllin egzekutiv.
- **Make**- mundëson që të ndërtoj project që përbëhet nga dy apo më shume skedarë

Opsioni Debug

- **Breakpoint** –tregon pikë para të cilës ndalon egzekutimi I programit, vendoset për qëllim debugimi
- **Add breakpoint...**-vendos pika para të cilës ndalon egzekutimi I programit

Opsioni Help

- **Contents**- tregon përmbajtjen e tabelës për ndihmë
- **Index**- tregon në ekran indeksi i ndihmës
- **Topic search**- paraqet në ekran temën e selektuar

PYETJE DHE DETYRA

1. Sqaro detyrat e sistemit operativ
2. Trego shembuj per detyra te softuerit sistemor dhe aplikativ në një system kompjuterik
3. Çka paraqet algoritmi, trego shembuj nga jeta e përditshme, sqaro hapat e zgjidhjes së problemit
4. Të shkruhet algoritmi i cili gjen shumën e dy kohëve të dhëna në orë dhe minuta.
5. Të shkruhet algoritmi i cili gjen dallimin e dy kohëve të dhëna në orë minuta dhe sekonda
6. Të shkruhet algoritmi i cili gjen shumën e dy thyesave të dhëna në formën $\frac{a}{b}$ dhe $\frac{c}{d}$.
7. Sqaro detyrat e gjuhëve programuese dhe ndarjen e gjuhëve programuese
8. Cilat janë përparësitë dhe mangësitë e gjuhës assembler ndaj gjuhëve të larta programuese
9. Në cilin grup të të dhënave nga tabela bëjnë pjesë të dhënat në vijim: emri, mbiemri, numri i ditarit, zgjidhjet e barazimit katror në matematikë, ngjyra e syve, trungu familjar, lëndët mësimore për vitin e dytë.
10. Trego se cili tip i ndryshores do ti përgjigjet të dhënave në vijim: numri i ditarit, emri, mbiemri, gjinia, mosha, data e lindjes.

11. Sqaroni dallimet në mes të kompajluesit dhe interpretuesit
12. Shkruaj algoritmin nga krijimi deri tek egzekutimi i një Paskal programi duke përdorur opsionet që ofron Turbo Pascal for Ėindoës (TPĖ).

Detyrat në vijim të zgjidhen duke përdorur TIK

- Të gjenden mënyrat e shprehjse së algoritmit
- Të sqarohen funksionet e kompjaluesit dhe interpretuesit

2.TIPET THEMELORE TË TË DHËNAVE

Kapitulli 2

Qëllimet e temës

- Njohja dhe përdorimi i tipeve të thjeshta të të dhënave (**cardinal integer,real, Boolean, char, string**);
- Të krijojë identifikator (emra) për ndryshore dhe konstante;
- Njohja e formës bazë të një programit në Pascal (titulli, pjesa e deklarimeve, pjesa e egzekutimit dhe komentet)
- Të përdorë tip cardinal dhe numër të plotë, real;
- Zbatimi notacionin e duhur eksponencial për paraqitjen e numrave real;
- Njohja dhe përdorimil ndryshoreve të tipit logjik,karakter dhe string;
- Të pranojë të dhëna nga tastatura(urdhëri **read** dhe **readln**);
- Të dërgojë të dhëna në ekran (urdhëri **write** dhe **writeln**);
- Të formatojë – editojë daljen (dërgimin në ekran) të të dhënave
- Të dallojë ndryshore dhe konstanta;

Njohuritë e nevojshme nga lëndët tjera:

- Matematika
- Informatika

2.1. ELEMENTET E GJUHËS PASCAL

Sikurse gjuhët natyrore edhe gjuha programuese Pascal e ka bashkësinë e simboleve (shenjave) të veta ose alfabetin e vet. Me simbolet e alfabetit krijohen sipas rregullave (sintaksës) së gjuhës programuese Pascal krijohen fjalët (konstruktionet e thjeshta) dhe urdhëra (konstruktionet e zgjeruara) në këtë gjuhë. Me kupitimin e konstrukcioneve të ndryshme në një gjuhë merret semantika e asaj gjuhe. Ndërsa alfabeti, fjalët, urdhërat, sintaksa etj, janë elementet e një gjuhe.

ALFABETI I GJUHËS PROGRAMUESE PASCAL

Në gjuhën Programuese Pascal përdoren këto simbole:

1. 26 shkronja të alfabetit latin
A,B,C,....,Z gjegjësisht a,b,c,.... z
2. 10 shifrat 0,1,2,3,4,5,6,7,8,9 për numra
3. Shenjat e veçanta
 - Shenjat për veprime aritmetike + - * /
 - Shenjat për krahasime <> =
 - Kllapa () [] { }
 - Shenja tjera si _ . , ; ^ , etj
 - Hapsira

Sikurse gjuhët natyrore duke i renditur simbolet e gjuhës PASCAL sipas rregullave sintaksore të saj fitohen konstruktionet e gjuhës.

FJALËT E GJUHËS PASCAL

Fjalët e gjuhës programuese pascal ndahen në dy grupe:

- **Fjalë të rezervuara** dhe
- **Identifikator**

Fjalët e rezervuara ose fjalë kyçe (ang. KEY WORDS) janë disa fjalë të gjuhës Pascal të cilat kur përdoren, duhet të shkruhen saktësisht ashtu siç janë përkufizuar. Këto fjalë kanë domethënie të caktuar për kompajlerin e Pascal-it dhe janë të ndërtuara duke u bazuar në fjalët e gjuhës angleze.

Lista me disa nga fjalët e rezervuara në Turbo Pascal

AND	ARRAY	BEGIN	CASE	CONST	DIV
DO	ELSE	END	FOR	FUNCTION	GOTO
IF	IN	MOD	NOT	OR	PROGRAM
RECORD	REPEAT	THEN	TO	TYPE	UNTIL
VAR	WHILE				

Identifikatorët ose emrat simbolik janë fjalë të cilat programuesi i përkufizon (i zgjedh) sipas rregullave të caktuara dhe të cilat përdoren në program. Me identifikatorë emërohen: programi, llojet(tipet) e ndryshme të të dhënave, ndryshoret, procedurat dhe funksionet.

Identifikatorët mund të ndërtohen si vargje të simboleve të caktuar nga alfabeti i gjuhës Pascal. Vargjet e tilla bëhen si kombinacione të shkronjave të alfabetit latin (A..Z, a..z), simbolit `_` dhe shifrave numerike 0 deri në 9. Identifikatori duhet të fillojë me shkronjë ose `_`. Identifikatorët nuk mund të emërohen me fjalë të rezervuara, me emra të konstanteve, funksioneve ose procedurave standarde, Gjithashtu identifikatori nuk guxon të përmbajë hapsirë. Gjuha programuese Pascal nuk dallon shkronjën e vogël nga e madhja. Fjalët **ëhile** dhe **ËHILE** kanë "kuptim" të njëjtë, si dhe **shuma dhe Shuma** nuk kanë asnjë dallim në kodin programues.

Në vijim po paraqesim disa emërtime të gabuara dhe të sakta për identifikator

Gabimisht	Saktë
32	Nr32
2x	DyX
Shuma e numrave	ShumaeNumrave
m/sek	mNeSek
Var	Ndryshore
CONST	Konstante
Barazi=	Barazi
X ²	XneFuqi2

2.2. STRUKTURA E NJË PASCAL PROGRAMI

Programet e shkruara në gjuhën programuese Pascal i përmbahen rregullave të caktuara të cilave duhet që ti përmbahemi gjatë shkruajes së një programi. Para se të studiojmë komponentët përbërëse veçmas të tregojmë strukturën e një programi në gjuhën programuese Pascal.

Një Pascal program elementar përbëhet nga këto pjesë:

- Emri i programit
- Urdhërat paraprocesorike
- Deklarimi i tipeve
- Deklarimi i konstanteve

- Deklarimi i ndryshoreve
- Blloku i programit kryesor
- Urdhërat dhe shprehjet brenda çdo bllokut
- Komentet

Por në përgjithësi çdo program në Pascal ka pjesën e titullimit, pjesën e deklarimeve dhe pjesën për egzekutim në radhitje strikte. Formatimi në vazhdim tregon sintaksën bazë të një Pascal programi.

```
program {Emri I programit}
uses {emrat e bibliotekave qe do ti perdorni te ndara me presje}
const{ deklarimi I konstanteve }
VAR{bloku I deklarimeve tw ndryshoreve}

BEGIN {Fillimi I bllokut programor kryesor}
...
END.{Fundi I bllokut programor kryesor}
```

Mirëpo një program funksional nuk është e domosdoshme të përmbajë të gjitha blloqet e sipërpërmendura por varësisht nga nevoja dhe kompleksiteti e programit që duhet shkruar.

PROGRAMI I PARË NË TURBO PASCAL

Të shkruajmë programin e parë në Pascal detyra e të cilit është të paraqesë në ekran tekstin **Miresevini ne Programim**.

Pasi të keni instaluar programin Turbo Pascal 1.5 dhe startoni prgramin nga ikona në ekran do të paraqitet Dritarja e gjuhës Turbo Pascal sikurse në figuren 6.

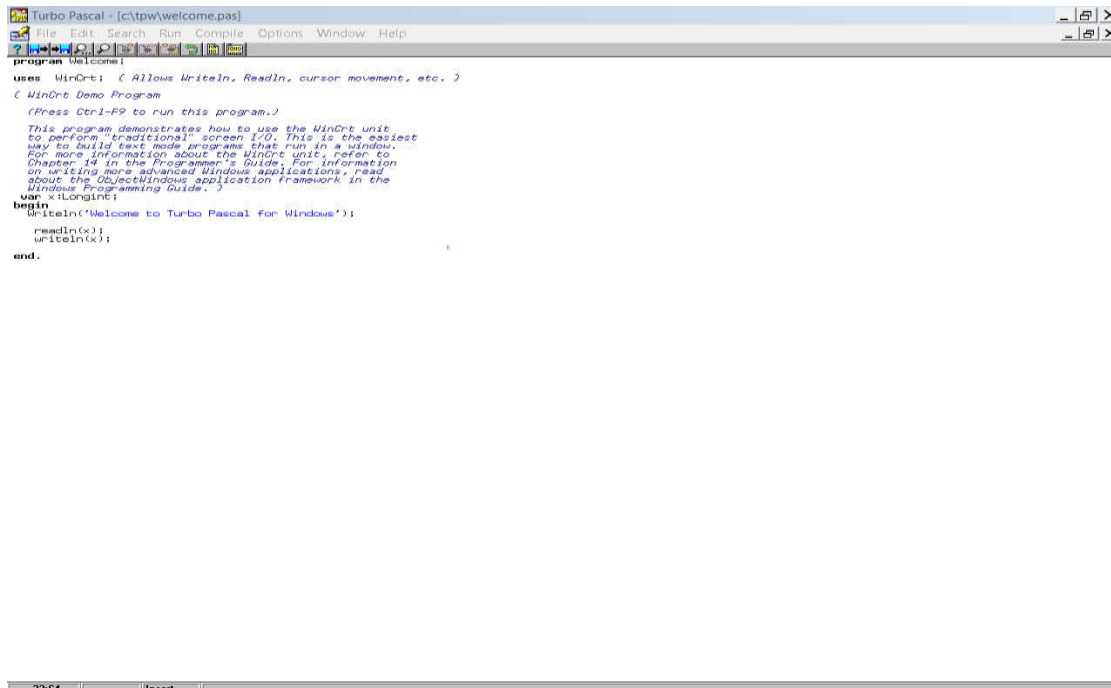


Figura 6 Dritarja fillestare në Turbo Pascal

Duke u bazuar në rregullat e gjuhës programore mund të shkruajmë programin e parë në Turbo Pascal. Dritarja mundëson që të shkruhet dhe editohet kodi i programit. Kodi i cili mundëson që të paraqesë në ekran tekstin “ Miresevini ne programim” do të duket si në figurën e dhënë.

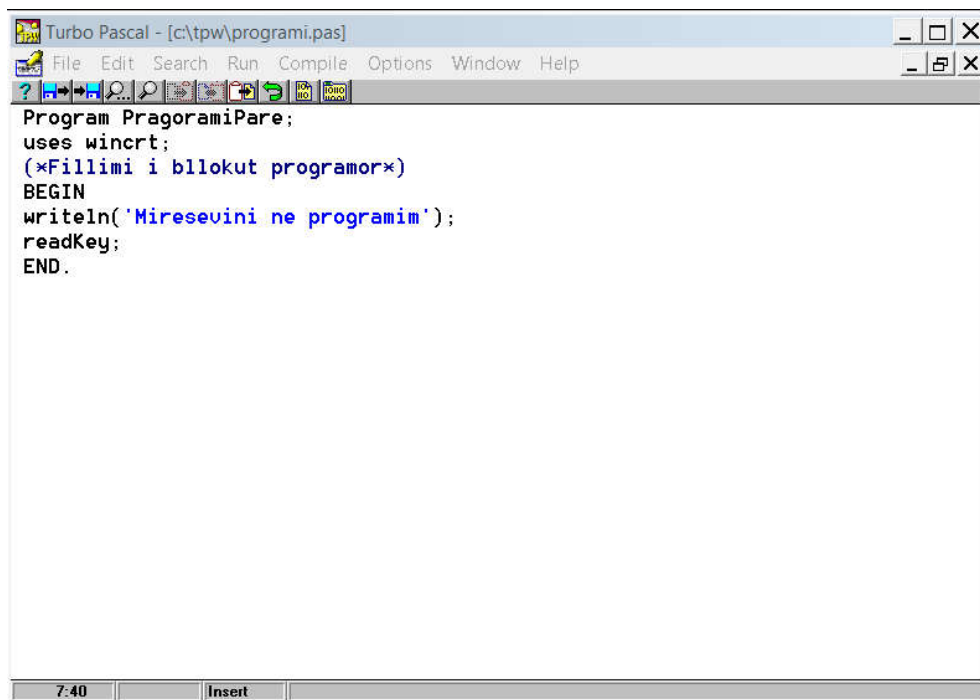


Figura 7 Programi i parë në Turbo Pascal

Për ruajtjen e programit e kodit programor në memorjen periferike përdoret opsioni File-> Save as.. sikurse në figurën 8. Zgjidhet emri i programit me prapashtesën .pas dhe vendi ku do të ruhet fajli sikurse është paraqitur në figurën 9.

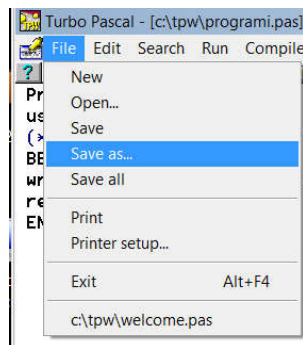


Figura 8 Ruajtja e kodit programor

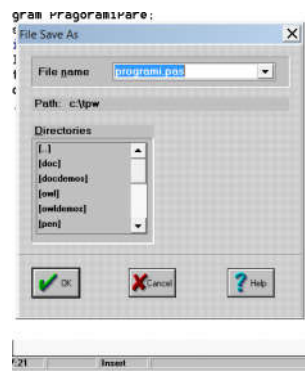


Figura 9 Ruajtja dhe emertimi e kodit programor në skedar

Për kompjalimin (pwrkthimin) e kodit programor nga gjuha pascal në gjuhën e makinës përdoret opsioni Compile nga menyuja sikurse në figurën 5 ose shkurtesa Alt+F9. Nëse programi është shkruar pa gabime sintaksore atëherë në ekran do të paraqitet mesazhi si në figurën 6.

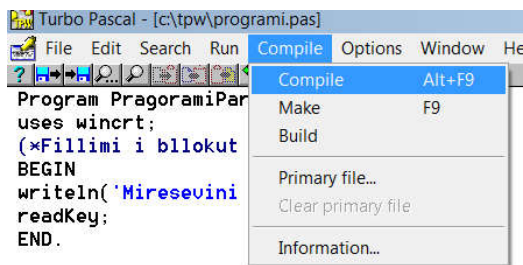


Figure 10 Kompajlimi i programit

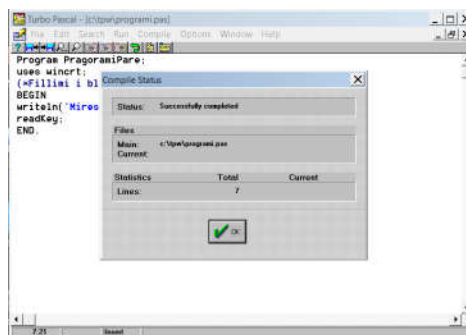


Figure 11 Mesazhi i kompajlimit të suksesshëm

Për ekzekutimin e programit përdoret opsioni Run nga menyuja rënëse **Run** ose shkurtesa Ctrl+F9, Figura 12.

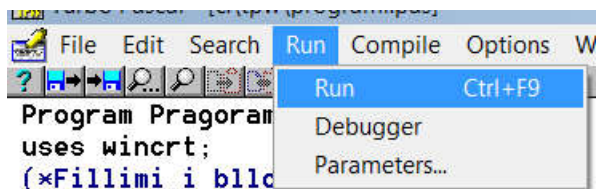


Figure 12 Ekzekutimi i programit

Në ekran do të paraqitet dritare e re që përmban mesazhin i dëshiruar nga programuesi.

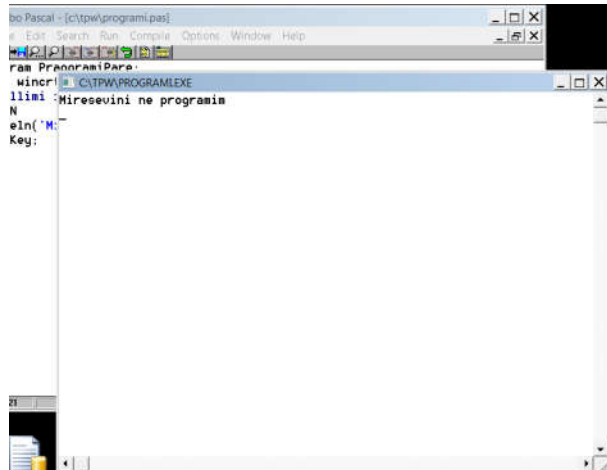


Figura 13 Rezultati nga ekzekutimi I programit të parë

Tani që programi u egzekutua me sukses të analizojmë kodin e shkruar nga programuesi.

- Rreshti i parë e programit **Program ProgramiPare;** tregon emrin e programit fjalën e rezervuar **Program** duhet që ta përmbajë çdo program, ndërsa identifikatori **ProgramiPare** është emri të cilin e ka zgjedhur programuesi për të emërtuar programin.
- Rreshti i dytë e programit **uses crt;** është urdhër paraprocesorikl cili urdhëron kompajleri të përfshijë njësine (bibliotekën) e gatshme e cila mundëson përdorimin e urdhërit për paraqitjen e të dhënave dhe tekstit në ekran
- Teksti i shkruar brenda shenjave (* *) është komentimi i programit dhe nuk lexohet nga kompajluesi dhe njihet si koment i programit
- Vijat programore të shkruara në mes të **BEGIN** dhe **END** janë blloku kryesor i programit, çdo bllok në pascal është i mbyllur me një **BEGIN** dhe **END**. Fjala e rezervuar **END** paraqitja e së cilës tregon fundin e programit pason nga një pikë (.) për dallim nga urdhërat tjerë që përfundojnë me pikëpresje(;
- **BEGIN** deklarimi në programin kryesor është rreshti ku fillon egzekutimi i urdhërave të programit
- Urdhëri **Writeln('Miresevini ne programim');** përdorë funksionin ëriteln të gjuhës Pascal i cili mesazhin në thonjëza të njëfishtë **Miresevini ne programim** do ta shfaq në ekran

- Urdhëri **readKey**; mundëson që ekran i rezultateve të programit të shfaqet derisa shfrytëzuesi të shtyp ndonjë tast. Është pjesë e **crt** bibliotekës.
- Urdhëri i fundit **END**. Përfundon programin

Komentet

Komentet shërbëjnë për të sqaruar ose përshkruar kodin, me qëllim që kodi i shkruar të jetë i qartë për të tjerë por edhe vetë autorin e kodit. Komentet shkruhen në guhë natyrore dhe duhet të jenë sa më të qarta deri në sqarimin e detaleve të kodit të shkruar me qëllim që pas një kohe vetë programuesi ose ata të cilët shohin kodin e programit të kuptojnë qëllimin dhe domethënësin e urdhërave të shkruar.

Komentet mund të vendosen në çdo pjesë të programit.

Komenti kufizohet me kllapat e mëdha si {ky tekst është koment }, i cili gjithashtu mund të shkruhet në kllapa të vogla me shenjën (*) sikurse (*ky tekst është koment *).

Komenti injorohet (nuk lexohet) nga përkthyesi (kompajluesi).

Shembull i mëposhtëm tregon komentimin e një programi duke sqaruar qëllimin e programit dhe çdo vijë të shënuar të tij, kemi përdorur dy llojet e komentimit.

```
{Progrmi i cili tregon ne ekran tekstin  
miresevini nw programim dhe kursorin e vendos ne rresht tw ri}  
  
Program ProgramiPare; (*emri i programit*);  
  
uses wincrt; {libraria e cila mundeson pwrdorimin e urdherave per paraqitje  
te teksit ne ekran}  
  
Begin (*fillimi i bllokut kryesor programor*)  
  
writeln('miresevini nw programim'); (*urdheri per paraqitjen e tekstit ne  
ekran*)  
  
readKey; (*urdheri per te lexuar vlere nga tastiera*)  
  
END. (*fundi i bllokut kryesor programor*)
```

2.3. NDRYSHORET, DEKLARIMI I TYRE

Tipet e të dhënave për një entitet në parim tregojnë tregojnë, kuptimin, kufizimin, vlerat e mundëshme, operacionet, funksionet dhe mënyrën e ruajtjes së tyre.

NDRYSHORET - ruajtja e të dhënave nga gjuhët programuese bëhet në memorje të rezervuar më parë ose në mënyrë dinamike gjatë proçit të egzekutimit të programit. Vlera e ruajtur në një adresë të caktuar të memorjes mund ndryshojë gjatë egzekutimit të programit. Për atë shkak një element i rëndësishëm i gjuhës programuese Pascal janë edhe *ndryshoret*.

Ndryshoret paraqesin edhe emra të cilat iu jipen pjeseve të vecanta të memorjes ashtuqë programi jonë të manipulojë me vlerën e ruajtur në atë pjesë të memorjes.

Për të kryer operacionet me ndryshoret në memorje gjuha programuese Pascal duhet ti organizojnë të dhënat duke ditur tipin e vlerës. Tipi i të dhënës është edhe karakteristika kryesore e saj. Tipet themelore të ndryshoreve në Turbo Paskal do ti paraqesim në tabelën në vazhdim.

Emri i tipit	Tipi vlerws sw të dhënës	Rangu	Shembuj
Boolean	Vlera logjike	True,False	TRUE, FALSE
Char	Karakter të vetëm		'A', 'E', '@', '6', 'n'
Byte	Numwr I plotw	0..255	56,139,201
Integer	numra të plotë	-32768-32767	3, 9652,-6354
LongInt	numra të plotë	-2147483648.. 2147483647	36589,-45879
Real	Numra dhjetor		3.14, 503.2
String	Përmban text	Deri në 255 karaktere	'Programim', 'Shkolla', '123'

Boolean (tip logji i të dhënës) Mund të ruhen vlera TRUE (1) dhe FALSE (0).

Char- ndryshorja në vete përmban karakter si 'a', 'B', '7', '\$'.

Byte - tipi i të dhënës përmban të dhënat numër i plotë në intervalin 0 deri në 255

Shortint- tipi i të dhënës përmban të dhënat numër i plotë në intervalin -128 deri në 127.

Integer- tipi i të dhënës përmban të dhënat numër i plotë në intervalin -32768 deri në 32767

Real- është tip I të dhënës numër me presje dhjetore në rang prej 3.4×10^{-38} deri në 3.4×10^{38} , rang vlen njësoj edhe për numrat negative. Vlerat ruhen me notacion shkencor, me mantis dhe eksponent.

String – tip I të dhënës varg karakteresh gjatësia maksimale 255 ku gjatësia e tij mund të specifikohet si e dhënë opsionale.

Vlen të theksohet se rang i vlerave varet gjithashtu nga kompajleri dhe sisemi operativ në të cilin përkthehet dhe egzekutohet programi.

Me të gjitha tipet e të dhënave operacionet dhe funksionet me tipet e të dhënave do të njihemi më detalisht në njësitë mësimore në njësitë në vazhdim.

Në kapitullin e mëparshëm i ndamë tipet e të dhënave. Në vazhdim do të sqarojmë deklarin e ndryshoreve, mënyrën e qasjes deri te ndryshoret.

Ndryshoret deklarohen pas bllokut të emrave të bibliotekave dhe para bllokut kryesor të programit.

Për deklarimin e ndryshoreve në gjuhën programuese Pascal pascal përdoret fjala e rezervuar **VAR** dhe më pas shkruhet identifikatori për emrin e ndryshores shenja e pikësimit dy pika (:), tipi i ndryshores dhe urdhëri mbaron me (;) pikëpresje.

```
VAR numri: INTEGER;
```

Me kodin e sipërm është deklaruar ndryshorja *numri* e tipit numër i plotë (ang. integer). Vlerat e mundshme të ndryshores *numri* janë nga -32768 deri në 32767. Përmes emrit të ndryshores i qasemi vlerës së ndryshores në memorje, e cila mund të sqarohet përmes figurës në vijim.

Emri i ndryshores	vlera e ndryshores
<i>numri</i>	-----

Me deklarimin e ndryshores të tipit numër i plotë përveç rezervimit të memorjes për ruajtje të vlerës numër i plotë vlera e memorjes inicializohet (jipet vlerë fillestare) zero.

Emri i ndryshores	vlera e ndryshores
<i>numri</i>	<u>000000000000</u>

Deklarimi i dy apo më tepër ndryshoreve të tipit të njejtë bëhet në mënyrën që ndryshoret e tipit të njejtë ndahen me presje (,) dhe deklarohet tipi i tyre.

```
VAR numri1,numri2,numri3: INTEGER;
```

Deklarimi i ndryshoreve të tipeve të ndryshme bëhet ashtu që ndryshoret mund të grupohen sipas tipit:

```
VAR numri1,numri2,numri3: INTEGER;  
    Shkronja1,shkronja2, shkronja3: CHAR;  
    Gjykimi:BOOLEAN;  
    Emri:STRING;
```

Ose forma e përgjithshme

```
VAR lista me ndryshore 1 :Tipi 1;  
    lista me ndryshore 2 :Tipi 2;  
    . . . . .  
    lista me ndryshore N :Tipi N;
```

2.4. INICIALIZIMI DHE PARAQITJA NË EKRAN I NDRYSHOREVE

Inicializimi i ndryshores nënkupton shënimin e vlerës fillestare në memorjen përkatëse të asaj ndryshore. Në Pascal inicializimi i ndryshores bëhet me urdhërin:

```
Ndryshorja:=shprehja;
```

ku kusht i domosdoshëm është që shprehja dhe ndryshorja të jenë të tipit të njejtë. Nëse ndryshorja është e tipit Integer (numër i plotë) edhe shprehja duhet të jetë Integer. Shprehjet janë konsruksione të cilat specifikojnë rregulla për llogaritjen e ndonjë vlere.

Shembulli në vazhdim deklaron dhe inicializon ndryshoret emri,mbiemri,nrDitarit, gjinia dhe mosha me vlerat e tipit të definuar më parë.

Fillimisht për çdo ndryshore duhet të zgjedhim tipin e përshtatshëm për të ruajtur vlerat përkatëse të tyre. Ndryshoret emri dhe mbiemri paraqesin varg karakteresh dhe duhet të ruhen si ndryshore të tipit String; Numri i ditarit paraqet ndryshore të tipit numër i plotë jo më i madh se 35 andaj ruhet si **byte**, duke deklaruar edhe moshën të njejtin tip. Ndërsa vlerat e ndryshores gjinia mund të jenë M dhe F dhe i përgjigjet tipi i të dhënës **char**.

```
Program Inicializimi;  
uses wincrt;
```

```
VAR emri,mbiemri:String;
nrDitarit,mosha:byte;
gjinia:char;
(*Fillimi i bllokut programor*)
BEGIN
emri:='Liridon'; (* Ndryshore emri i jepet vlera Liridon*)
mbiemri:='Besimi'; (* Ndryshore mbiemri i jepet vlera Besimi*)
gjinia:='M';      (* Ndryshore gjinia i jepet vlera M*)
nrDitarit:=7;     (* Ndryshore nrDitarit i jepet vlera 7*)
mosha:=16;        (* Ndryshore mosha i jepet vlera 16*)
readKey;
END.
```

Vlerat e tipit string dhe char shkruhen nw thonjwza.

PARAQITJA NW EKRAN e ndryshoreve bwhet me urdhwrat `WRITE(ndryshorja);` ose `WRITELN(ndryshorja);`. Ose nëse kemi dy apo më tepër ndryshoret ndahen me presje: `WRITE(ndryshorja1,ndryshorja2,...,ndryshorjaN);`

`WRITELN(ndryshorja1,ndryshorja2,...,ndryshorjaN);`

Dallimi në mes të **WRITE** dhe **WRITELN** është se **WRITE** tregon vlerën e shprehjes në ekran dhe vendos kursorin menjëher pas vlerës së paraqitur në ekran, ndërsa **WRITELN** paraqet vlerat në ekran dhe çvendos kursorin në rresht të ri. Shembull urdhërat në vazhdim.

Nga shembulli I mëparshëm të paraqiten vlerat e ndryshoreve në ekran.

```
Program Inicializimi;
uses wincrt;
VAR emri,mbiemri:String;
nrDitarit,mosha:byte;
gjinia:char;
(*Fillimi i bllokut programor*)
BEGIN
emri:='Liridon';
mbiemri:='Besimi';
gjinia:='M';
nrDitarit:=7;
mosha:=16;
WRITELN('Te dhenat e nxenesit');
WRITE('Numri i ditarit:');
WRITELN(nrDitarit);
WRITE('Emri:');
WRITELN(emri);
WRITE('Mbiemri:');
WRITELN(mbiemri);
WRITE('Mosha');
WRITELN(mosha);
WRITE('Gjinia');
WRITELN(gjinia);
readKey;
```

END.

Pas kompajlimit dhe egzekutimit të programit në ekran do të paraqitet dritarja në të cilën tregohen të dhënat e nxënësit gjegjësisht vlerat e ndryshoreve.

Te dhenat e nxenesit

Numri i ditarit:7

Emri: Liridon

Mbiemri: Besimi

Mosha: 16

gjinia: M

Përmes urdhërit WRITE/WRITELN mund të paraqesim në ekran vetëm me një urdhër më shumë ndryshore ose konstanta tekstuale të cilat në mes veti janë të ndara me presje.

```
WRITELN('Te dhenat e nxenesit');  
WRITELN('Numri i ditarit:',nrDitarit);  
WRITE('Emri:',emri);  
WRITE('Mbiemri:',mbiemri);  
WRITE('Mosha',mosha);  
WRITE('Gjinia',gjinia);
```

Sigurisht se mënyra e dytë e paraqitjes, duke ndarë ndryshoret nga konstantat tekstuale me presje (,) mundëson që me më pak kod të shkruar të zgjidhim të njejtin problem.

2.5. INICIALIZIMI I NDRYSHOREVE PËRMES TASTIERËS

Deri tani arritëm të mësojmë deklarimin dhe inicializimin ndryshores përmes kodit programor, por e dimës se futja e të dhënave në kompjuter për tu përpunuar zakonisht bëhet përmes tastierës.

Urdhërat që shërbejnë për ti dhënë ndryshores vlerë përmes tastierës nga përdoruesi gjatë egzekutimit të programit janë urdhërat READ dhe READLN sintaksa e përdorimit të tyre është:

```
READ(ndryshorja);  
READLN(ndryshorja);
```

Njësoj sikurse urdhëri për paraqitjen e të dhënave në ekran edhe urdhëri readln mund të bëjë leximin e më shumë ndryshoreve me një urdhër;

```
READLN(ndryshorja_1,ndryshorja_2,...,ndryshorja_n);  
READ(ndryshorja_1,ndryshorja_2,...,ndryshorja_n);
```

Urdhërat READ/READLN presin nga shfrytëzuesi të futen vlerat e ndryshoreve përmes tastaturës dhe me shtypjen e tasit **Enter ndryshoret inicializohen (marrin vlerat) të dhëna me tastierë në mënyrë përkatëse për çdo ndryshore.**

Dallimi ne mes read dhe readln

readln(a,b,c,d);

readln(a,b);

writeln(a,b,c,d);

read(a,b,c,d);

read(a,b);

writeln(a,b,c,d);

readln;

readln;

Shembull i leximit të ndryshores a përmes tastierës:

```
Program Lexo;  
Uses wincrt;  
var a: integer;  
BEGIN  
WRITE('Shkruaj vleren e ndryshores a: ');  
READLN(a);  
WRITELN('Ndryshorja a përmban vleren',a);  
ReadKey;  
END.
```

Ekzekutimi i kodit programor të mësipërm do të mundësojë që shfrytëzuesi të mund të fusë vlerë për ndryshoren **a** përmes tastierës sikurse në fig. vijuese:



Figura 14 Urdhëri READLN

Kursori do të venoset pas tekstit të treguar në ekran dhe pret që ti jipet **vlerë e tipit numër i plotë** përmes tastierës dhe vlerën e dhënë të inicializohet ndryshorja a, e tipit numër i plotë.

Dallimi në mes të READ dhe READLN është se READ mund të lexojë edhe më shumë vlera të ndara me hapsirë nga një rresht, ndërsa READLN lexon vetëm vlerën për ndryshoren e parë dhe vlerat tejra të atij rreshti nuk i merr parasysh, por mundëson të shkruhen në rreshtin e ri.

Shembull i përdorimit të urdhërit read dhe readln mund ta praktikojmë duke rizgjidhur shembullin paraprak të inicializimit të vlerave për të dhënat e nxënësit, duke bërë inicializimin e ndryshoreve përmes tastierës.

```
Program Inicializimi;
uses wincrt;
VAR emri,mbiemri:String;
nrDitarit,mosha:byte;
gjinia:char;
(*Fillimi i bllokut programor*)
BEGIN
emri:='Liridon';
mbiemri:='Besimi';
gjinia:='M';
nrDitarit:=7;
mosha:=16;
WRITELN('leximi i te dhenave per nxenesin permes tastieres');
WRITE('Numri i ditarit:');
READLN(nrDitarit);
WRITE('Emri:');
READLN(emri);
WRITE('Mbiemri:');
READLN(mbiemri);
WRITE('Mosha');
READLN(mosha);
WRITE('Gjinia');
READLN(gjinia);
readKey;
END.
```

2.6. DEKLARIMI I TIPEVE

Deklarimi i tipeve mundëson emertimin e tipit të ndryshores sipas dëshirës së programuesit. Me deklarimin e tipit mund të bëhet deklarimi i ndryshores me tipin e definuar nga vetë programuesi. Deklarimi i tipit mund të bëhet si: në bllokun e deklarimit të tipeve me fjalën e rezervuar TYPE. Sintaksa e deklarimit të tipit është.

```
TYPE identifikatori1,identifikatori2,...,ideftifikatoriN = TIPI;
```

Ndërsa deklarimi i ndryshores bëhet:

```
Var ndryshorja1, ndryshorja2, ... ,ndryshorjaN :identifikatori1;
```

Shembull I deklarimit të tipeve;

```
Type mosha,gjatesia,pesha=byte;  
Tekst=string;  
Gjykim=Boolean;  
Var m:mosha; L:gjatwsia;  
p:Gjykim;
```

2.7.1. KONSTANTET

Konstatnet paraqesin entitete të cilat nuk ndryshojnë gjatë egzekutimit të programit. Në Pascal dallojmë dy tipe të konstanteve

- Konstantë e emëruar
- Konstantë e definuar

Konstata e definuar është vlerë karakter numër ose tekst dhe definohet sikurse ndryshoret por në program nuk ndryshon vlera e saj. Emri dhe vlera e konstanës deklarohen para bllokut VAR,Sintaksa e deklarimit të konstantës është si në vijim.

```
Const Pi=3.14;  
  
Const million=1000000;
```

Disa konstanta janë të definuara në Pascal si identifikator standard. Një konstantë e tillë është konstanata MAXINT, vlera e së cilës është vlera maksimale e ndryshores së tipit Integer. Për ta gjetur vlerën maksimale të Integerit për kompajklerin tuaj përdoret urdhëri.

```
WRITELN(MAXINT);
```

Vlera e të cilit është 32767 (2 bajt).

Shembull i përdorimit të konstantës PI në vazhdim do të përdorim për caktimin e perimetrit dhe syprinës së rethit.

```
PROGRAM Konstantet;  
CONST Pi = 3.14159;  
VAR rrezja, Perimetri :REAL;  
BEGIN  
    rrezja:= 4.9;
```

```
Perimetri :=2*Pi*rrezja;  
WRITELN('Perimetri=', Perimetri)  
  READKEY;  
END.
```

2.7. TË DHËNAT LOGJIKE OPERATORËT LOGJIKE PARAQITJA E TYRE NË EKRAN

Ndryshoret e tipit Boolean deklarohen me fjalën e rezervuar Boolean në formën:

```
Var p,q:boolean;
```

Vlerat e mundshme të ndryshoreve të tipit logjik janë True (e saktë) dhe False (e pasaktë). Operacione themlore me të dhënat e tipit logjik në Pascal janë: NOT, AND, OR dhe XOR; Rezultatet nga operatorët logjik janë ndryshore të tipit logjik (BOOLEAN).

Inicializimi i ndryshoreve të tipit logjik bëhet përmes operatorit := në formën **p:=shprehja**.

Ndryshorja **p** dhe **shprehja** janë e tipit logjik. p:=true; ose p:=false;. E rëndësishme të theksohet se ndryshoret e tipit logjik nuk mund të inicializohen përmes urdhërit READLN;

NOT- negacion, AND- konjunksion, OR –disjunksion dhe XOR-disjunksion ekskluziv.

Programi në vijim do të tregojë shembuj me operacionet themelore logjike në ndryshore e tipit logjik.

```
Programi VleratLogjike;  
Uses wincrt;  
Var p,q,r: boolean;  
Begin  
  p:=true;  
  q:=false;  
  writeln('p=',p);  
  writeln('Not p=',not p);  
  writeln('q AND q=', p AND q);  
  writeln('q OR q=', p OR q);  
  writeln('q XOR q=', p XOR q);  
end.
```

Ushtrim: Të shtypet në ekran tabela e vërtetësive për operacionet themelore me të dhënat e tipit logjik.

Pascal mundëson edhe llogaritjen e shprehjeve më komplekse logjike duke përdorur kllapat () dhe pasur parasysh prioritetin e operatorëve shembull i tillë është edhe llogaritja e shprehjeve logjike ku p,q dhe r janë ndryshore të tipit logjik:

$p \text{ AND } q \text{ OR } (\text{NOT } p)$

$\text{NOT } p \text{ and } (\text{NOT } q \text{ XOR } r)$

$p \text{ OR } q \text{ and } r$

Në gjuhën programuese Pascal prioriteti i operatorëve logjik është sikur në tabelë

Prioriteti	Operatori
Më i lart	NOT
	AND
Më i ulët	OR,XOR

Table 1 Prioriteti i operatorëve logjik

Prioritet më i lartë ose llogaritjet e para, bëhen të shprehjeve në kllapa, pastaj vlen prioriteti i operatorëve nga tabela sikurse në Matematikë.

Detyrë: Të njehsohet vlera e shprehjes: $p \text{ OR } \text{NOT } q \text{ AND } \text{NOT } (p \text{ AND } r)$ për:

- a) $p=\text{True}, q=\text{True}, r=\text{False}$
- b) $p=\text{False}, q=\text{False}, r=\text{True}$
- c) $p=\text{False}, q=\text{True}, r=\text{False}$

2.8. TË DHËNAT NUMËR I PLOTË, OPERATORËT

Për të ruajtur numra të plotë gjuha programuese Pascal mundëson tipe të ndryshme të ndryshoreve të cilat në mes veti dallohen nga rangu i mundshëm i vlerave ose numri i bitëve që përdoren për të ruajtur ato vlera. Numrat e plotë në Pascal mund të ruhen në ndryshoret e tipit:

Tipi	Rangu i vlerave	Numri i bitëve
Integer	-32768..32767	16
ShortInt	-128..127	8
Byte	0..255	8
Longint	-2147483648.. 2147483647	32

Table 2 Domeni i vlerave për ndryshoret numër i plotë

Në implementime të ndryshme të gjuhës programuese Pascal ndryshoret mund të kenë domen të ndryshëm nga domeni i dhënë në tabelë.

Deklarimi i ndryshoreve bëhet në bllokun **VAR** të programit:

VAR

```
x,y,z: INTEGER;
a,b: BYTE;
l,v: LONGINT;
s: SHORTINT;
```

Me daklarimin e dhënë në memorje rezervohen vende për tri ndryshore(x,y,z) të tipit INTEGER, dy ndryshore(a,b) të tipit BYTE, dy ndryshore(l,v) të tipit LONGINT, dhe një ndryshore (s) e tipit SHORTINT.

Operacionet aritmetike me numra të plotë në Pascal janë operacionet aritmetike si shuma, ndryshimi, prodhimi herësi dhe mbetja për numrat e plotë implementohen përmes simboleve në tabelën vijuese:

Operatori	Përshkrimi
+	Shuma e numrave
-	Ndryshimi i numrave
*	Prodhimi i numrave
/	Herësi i numrave
DIV	Pjesa e plotë gjatë pjestimit
MOD	Mbetja gjatë pjestimit

Table 3 Operatorët aritmetik

Sintaksa e përdorimit të operatorëve aritmetik është:

```
Ndryshorja1 operatori Ndryshorja2 ose
Shprehja1 operatori Shprehja2
```

Shembull

```
x + y
a * b
9 MOD b
x / 3
15 div 4
```

Gjatë përdorimit të operatorëve DIV, MOD dhe / duhet pasur kujdes që operandi ose shprehja e dytë të ketë vlerë të ndryshme nga zero (pjestimi me 0 nuk është i lejuar) .

Paraqitja e numrave në ekran bëhet me urdhërin `ëriteln` ose `ërite` që mund që përdoren për të shfaqur ndryshore, numra ose shprehje.

Rezultati nga operatori / (herësi) është i tipit numër dhjetor ndërsa rezultati nga operatorë tjerë është numër i plotë.

Programi i cili ilustron operatorët aritmetik dhe paraqitjen e numrave të plotë në ekran.

```
PROGRAM OperatorwtAritmetik;  
Uses wincrt;  
BEGIN  
    WRITELN(123);  
    WRITELN(5+4);  
    WRITELN(7-4);  
    WRITELN(3 * 4);  
    WRITELN(8 / 2);  
    WRITELN(7 DIV 2);  
    WRITELN(10 MOD 3);  
    READKEY;  
END.
```

Urdhëri WRITELN(123); në ekran do të paraqesë në ekran numrin në kllapa gjegjësisht numrin 123,

Urdhëri WRITELN(5+4); cakton shumën e numrave 5 dhe 4, dhe në ekran paraqet numrin 9;

Urdhëri WRITELN(7-4); Cakton ndryshimin e numrave 7 dhe 4 dhe në ekran paraqet numrin 3.

Rezultatet nga urdhërat në vijim të programit janë 12, 2,3 dhe 1 në mënyrë përkatëse.

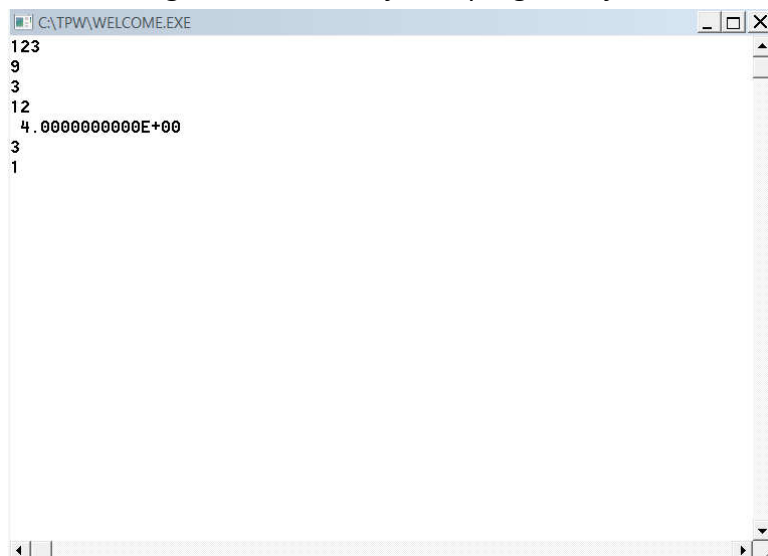


Figura 15 Operatorët aritmetik dhe paraqitja e numrave në ekran

Inicilaizimi i ndryshoreve numwr i plotw mund tw bwhet pwrmes kodit programor pwrmes urdhwrit.

Ndryshorja:=shprehja;

Ku ndryshorja dhe shprehja janw numwr i plotw.

Shembull:

```
x:=y*z+4;
l:=2*v+3;
a:= 2*b+12;
```

Inicialiaimi përme tastierës së ndryshoreve numër i plotë bëhet përmes urdhërave READ/READLN;

Kodi i programit i cili ilustron inicializimin e ndryshoreve përmes tastierës është:

```
PROGRAM Inicializimi;
Uses wincrt;
VAR
    A: INTEGER;
    B: BYTE;
BEGIN
WRITE( 'SHKRUAJ VLEREN PWR NDRYSHOREN A ');
READLN(A);
WRITE( 'SHKRUAJ VLEREN PWR NDRYSHOREN B ');
READLN(B);
WRITELN(VLERAT E NDRYSHOREVE JANE');
WRITELN( 'A= ',A,' B=', B);
READKEY;
END.
```

Gjatë egzekutimit të programit do të fitohet dritarja:

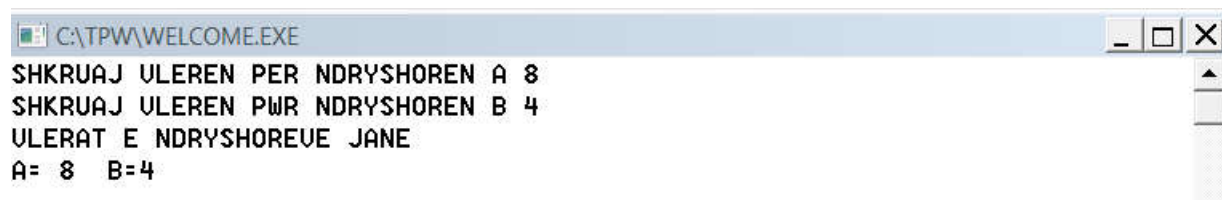


Figure 16 Inicializimi i ndryshoreve

Pascal mundëson edhe llogaritjen e shprehjeve më komplekse duke përdorur kllapat () dhe pasur parasysh prioritetin e operatorëve shembull i tillë është edhe llogaritja e shprehjeve ku x,y,z janë ndryshore të tipit Integer:

A:=x-y*75+42*(x-z)

B:=x+35 DIV Y -Z mod 3

$$C := 45 - 2 * x + 4 * y$$

Në pascal prioritetin e operatorëve e tregon tabela:

Prioriteti	Operatori
Më i lartë	*, /, DIV, MOD
Më i ulët	+, -

Table 4 Prioriteti i operatorëve

Nëse në një shprehje aritmetike kemi kllapa atëherë prioriteti i operatorëve ndryshon; së pari llogariten shprehjet brenda kllapave. Në shprehjet aritmetike në Pascal lejohet përdorimi vetëm i kllapave të vogla (). Vlejnë rregullat sikurse na matematikë, së pari njehsohet shprehja brenda kllapave. Rregullat për llogaritje të shprehjeve aritmetike janë ekuivalente me rregullat në aritmetike.

Duke shfrytëzuar prioritetin e operatorëve në Pascal të llogaritet rezultati i shprehjeve për $x=7, y=4$ dhe $z=-2$

- i. $A := x - y * 5 + 2 * (x - z)$
- ii. $B = x + 35 \text{ DIV } Y - Z \text{ mod } 3$
- iii. $C := 45 - 2 * x + 4 * y$

Zgjidhja:

- i. $A := x - y * 5 + 2 * (x - z) = 7 - 4 * 5 + 2 * (7 - (-2)) = 7 - 20 + 2 * (7 + 2) = -13 + 2 * 9 = -13 + 18 = 5$
- ii. $B = x + 35 \text{ DIV } Y - Z \text{ mod } 3 = 7 + 35 \text{ DIV } 4 - (-2) \text{ MOD } 3 = 7 + 8 - (-2) = 15 + 2 = 17$
- iii. $C := 45 - 2 * x + 4 * y = 45 - 2 * 7 + 4 * 4 = 45 - 14 + 16 = 31 + 16 = 47$

Shembull: Të shkruhet shprehja në Pascal e cila është ekuivalente me shprehjet aritmetike.

- i. $x = \frac{a+b}{c} - 2a$
- ii. $R = \frac{R1 \cdot R2}{R1 + R2}$
- iii. $X = A^2 - B^2$
- iv. $z = \frac{(a-b)c}{2a} - 4(a - 2b)$

Shprehja aritmetike (i) tregon se ndryshorja x merr vlerën e shprehjes së dhënë në anën e djathtë.

Atëherë shprehja në gjuhën programuese Pascal do të ishte në formën:

$$x := (a+b)/c - 2*a;$$

Shprehja (ii) ku llogaritet rezistance ekuivalente e dy rezistorëve të lidhur paralel në gjuhën programuese Pascal shkruhet:

$$R:=R1*R2/(R1+R2);$$

Urdhëri në gjuhën programuese Pascal i cili llogarit shprehjen (iii) mund të shkruhet në formën:

$$X:=A*A-B*B$$

Urdhëri në gjuhën programuese Pascal i cili llogarit shprehjen (iv) mund të shkruhet në formën:

$$Z:=(a-b)*c/(3*a)-4(a-2*b);$$

FORMATIMI I NDRYSHOREVE

Për të pasur një paraqitje më të lexueshme në ekran të të dhënave bëhet formatimi në urdhërin `WRITELN` nëpërmjet “rrafshimit” të tekstit dhe të dhënave.

Formatimi i të dhënave për paraqitjen e tyre në ekran bëhet ashtu që pas **ndryshore** vendosen (:) dhe jipet vlera se sa hapsira do të rezervohen për atë ndryshore. Ndryshorja rreshtohet nga e djathta aq vende sa është “komanduar” nga numri i plotë pas ndryshore. Nëse ndryshorja që shkruhet përmaban më shumë shifra se sa numri i hapsirave që rezervohen atëherë numri i hapsirave nuk do të mirret parasysh dhe ndryshorja do të shtypet e plotë pa lënë hapsirë të zbrazët para ndryshore.

Sintaksa (forma) e formatimit të ndryshoreve është:

```
WriteLn(ndryshorja:numriiVendeve);
```

Ilustrim i formatimit të numrave të plotë në Paskal:

```
PROGRAM shembulli;  
Uses wincrt;  
VAR i:INTEGER;  
BEGIN  
i:=234;  
WRITELN(i);  
WRITELN(i:5);  
WRITELN(i:6);  
READKEY;  
END.
```

Në ekran do të paraqesw:



2.8.1. FUNKSIONET STANDARDE ME NUMRA TË PLOTW

Gjatë programimit hasim me probleme të ndryshme në të cilat duhet llogaritje më të komplikua se operacionet themelore aritmetike (+,-,*,/, DIV dhe MOD). Nëse duhet të llogaritet vlera absolute, katrori, kubi, rrënja katrore dhe funksione tjera të cilat përpunojnë numra të plotë; gjuha programuese Turbo Pascal mundëson shfrytëzimin e funksioneve të definuara të cilat mundësojnë llogaritje të vlerave për funksione të ndryshme.

Funksionet të cilat përdoren më së shpeshti gjatë programimit dhe janë të përfshira në Pascal quhen funksione standard. Këto funksione janë pjesë përbërëse e Gjuhës programuese Pascal dhe përdoren gjatë shkruarjes së programit pa pasur nevojë që ti deklararojmë ose definojmë sjelljen e tyre.

Funksionet e shfrytëzohen(thirren) përmes emrit të tyre dhe parametrave të funksionit në kllapa. Forma e përgjithshme e thirrjes së një funksioni është

EmriFunksionit(arg1, arg2,...,argN);

Detyra e funksionit është që të llogarisë dhe kthejë vlerë e cila kërkohet nga programuesi. Në shembullin e njësisë mësimore të mëprashme gjatë llogaritjes se shprehjes: $X = A^2 - B^2$ për të shkruar urdhërin në gjuhën programuese Pascal e shkruam në formën: $X := A * A - B * B$, edhe pse rezultati gjatë llogaritjes është i saktë kodi i programit përmes funksionit bëhet më i kuptueshëm për lexuesin. Gjatë programimit mund të hasim në operacione si rrënja katrore, njehsimi i fuqisë do të ishte shumë më e vështirë programimi i operacioneve të tilla.

Libraria me funksione standarde mundëson përdorimin e funksioneve të cilat llogarisin vlerat e dëshiruara nga programuesi. Shprehjen 4^2 në gjuhën programuese Pascal mund ta shkruajmë si: `sqr(4)`; Emri i funksionit është shkurtesa **SQR**, funksioni ka vetëm një parametër (numër i plotë) 4, llogarit vlerën $4^2 = 16$ dhe kthen vlerë 16 numër i plotë.

Kodi në vijim do të ilustrojë përdorimin e funksionit SQR.

```
PROGRAM FunksioniKatror;  
Uses wincrt;
```

```

VAR
  X,Y: integer;
BEGIN
  X:= 6;
  Y:=SQR(x);
  WRITELN('vlera e Y eshte ',Y);
  WRITELN('SQR(4)= ',SQR(4));
  READKEY;
END.

```

Rezultat nga egzekutimi i programit të dhënë është:



Figure 17 Rezultati nga ilustrimi i funksionit SQR

Gjatë përdorimit të funksionit duhet të pasur kujdes në domenin e vlerave të tipit të vlerës kthyesë. Nëse duhet të gjejmë katrorin e numrit 200 atëherë ndryshorja e tipit INTEGER nuk mjafton për të ruajtur vlerën $40000 = \text{SQR}(200)$ (shiko tabelën x Domeni i vlerave për ndryshoret numëri plotë). Testoni funksionin SQR ku vlera e parametrës është

- i. 200
- ii. 2000
- iii. 20000

Në tabelën vijimë japim disa nga funksionet e definuara në librarinë standarde të gjuhës programuese Pascal ku parametri dhe rezultati kthehen si funksionit së bashku me numrin dhe plotë:

Funksioni	Përshkrimi
SQR(x)	x^2
ABS(x)	x vlera absolute e ndryshores x
RANDOM(X)	numër i rastësishëm në intervalin [0,x)
SUCC(x)	Pasardhësi i numrit x
PRED(x)	Paraardhësi i numrit x
ORD(x)	Numri rendor i x

Table 5 Funksionet standarde me parametër dhe rezultat numër i plotë

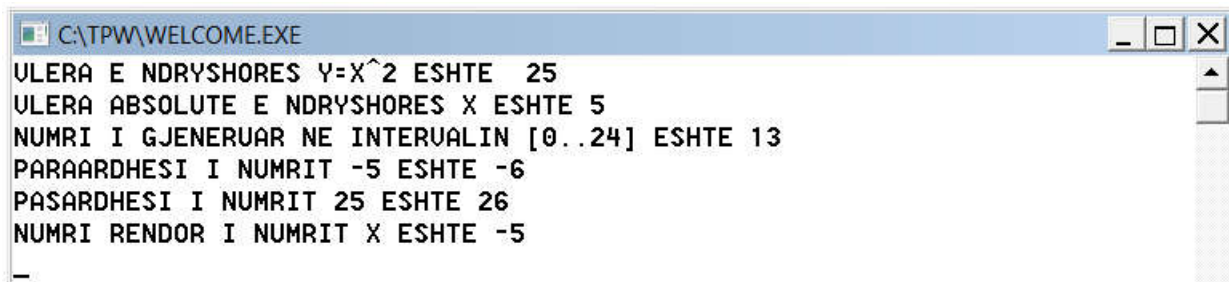
Kodi i programit i cili ilustron përdorimin e funksioneve nga tabela x:

```

PROGRAM FunkshonetStandarde;
Uses wincrt;
VAR X, Y:INTEGER;
BEGIN
    X:=-5;
    Y:=SQR(X);
    RANDOMIZE;
    WRITELN('VLERA E NDRYSHORES Y=X^2 ESHTË ',Y);
    WRITELN('VLERA ABSOLUTE E NDRYSHORES X ESHTË ',ABS(X));
    WRITELN('NUMRI I GJENERUAR NE INTERVALIN [0..'Y-1,'] ESHTË ',
RANDOM(Y));
    WRITELN('PARAARDHESI I NUMRIT ', X,' ESHTË ', PRED(X));
    WRITELN('PASARDHESI I NUMRIT ', Y,' ESHTË ', SUCC(Y));
    WRITELN('NUMRI RENDOR I NUMRIT X ESHTË ', ORD(X));
    READKEY;
END.

```

Rezultati nga egzekutimi I programit është:



```

C:\TPW\WELCOME.EXE
VLERA E NDRYSHORES Y=X^2 ESHTË 25
VLERA ABSOLUTE E NDRYSHORES X ESHTË 5
NUMRI I GJENERUAR NE INTERVALIN [0..24] ESHTË 13
PARAARDHESI I NUMRIT -5 ESHTË -6
PASARDHESI I NUMRIT 25 ESHTË 26
NUMRI RENDOR I NUMRIT X ESHTË -5

```

Figura x. Funkshonet me numra tw plotw

Për ta përdorur funksionin RANDOM duhet të shkruajmë urdhërin **RANDOMIZE**;

Funksionet me numrat të plotë rezultat i të cilat japin vlera logjike;

Funksioni	Pershkrim	Shembull
ODD(x)	Numri x është ose jo numër tek	ODD(1) {kthen vlerën TRUE} ODD(2) {kthen vlerën FALSE}

```

PROGRAM shembulli;
Uses wincrt;
VAR
    Numri:INTEGER;
BEGIN
    numri:=7;
    WRITELN(ODD(numri));
    Numri:=10;
    WRITELN(ODD(numri));
    READKEY;
END.

```




Figuri 18 Përdorimi i funksinit ODD

Lista e funksioneve të cilat parametër kanë numër të plotë ndërsa tipi i rezultatit që japin është numër real është dhënë në tabelën në vijim.

Funksioni	Përshkrimi
SQRT(x)	\sqrt{x}
SIN(x)	Sin(x)
COS(x)	Cos(x)
TG(x)	Tg(x)
CTG(x)	Ctg(x)
LN(x)	Log(x)
EXP(x)	e^x

Në shprehjet aritmetike, funksionet standard aritmetike mund të paraqiten edhe si argumente të funksioneve tjera. Për shembull

$$z = |x^2 - y^2|$$

në Paskal shkruhet si

`z=Abs (Sqr (x) -Sqr (y)) ;`

Ushtrime:

- Shkruaj vlerat e ndryshoreve të cilat i mund të ruhen në tipit e ndryshoreve { -15,124, -235,265, -32145,47234}
 - BYTE _____
 - INTEGER _____
 - LONGINT _____
 - SHORTINT _____
- Cili është tipi dhe vlera e rezultatit të shprehjeve;
 - $25 \bmod 4 * 3$
 - $75 * (2 + 9 \text{ DIV } 4)$

- c. $12+(35*2+3/2)$
- d. $36 \bmod 7 \div 3*2$
- 3. Të llogaritet vlera e shprehjes nëse $X=12$, $y=4$ DHE $Z=12$
 - a) $X \bmod Y \div (3 \bmod X)$
 - b) $X*Y \bmod 5+4$
 - c) $3*X+Z-Y \div 3$
- 4. Të shkruhen pjesa e kodit programor në Turbo Pascal të cilat njehsojnë:
 - a) Sipërfaqen e rrethit $S=r^2\pi$
 - b) Perimetrin e rrethit $P= 2r\pi$
 - c) Sipërfaqen e trekëndëshit nëse janë dhënë gjatësitë e brinjëve të tij përmes formulës së Heronit: $S_{\text{iperfaqja}}=\sqrt{s(s-a)(s-b)(s-c)}$ ku $s=\frac{a+b+c}{2}$
- 5. Të shkruhen në gjuhën programues Pascal shprehjet aritmetike:
 - a) $\left(\frac{x+4}{3-2y}\right)\left(\frac{2+x}{3}\right)$
 - b) $\frac{(a-b)}{c} + \frac{c(3+b)}{3a}$
- 6. Të llogaritet vlera e shprehjeve nëse $X=7$, $Y=4$:
 - a) $\text{SQR}(\text{PRED}(X)-\text{SUCC}(Y))/\text{SQRT}(\text{SUCC}(X)-\text{PRED}(Y));$
 - b) $\text{ODD}(\text{SQR}(X)-\text{SUCC}(\text{SQR}(Y)));$
- 7. Të caktohet tipi dhe vlera e rezultatit të shprehjes: $\text{chr}(\text{ord}(c))$ nëse c është ndryshore e tipit char dhe vlera e saj është 'd'.

2.9. NDRYSHORET E TIPIT NUMËR DHJETOR

Për ruajtjen e vlerave numër dhjetor si vlera 3.14, numri e 2.81, nota mesatre 4.68 gjuha programuese Pascal përdorë tipin e ndryshoreve REAL. Rangu i vlerave të ndryshoreve REAL ndryshon edhe nga platforma në të cilën egzekutohet programi. Për ruajtjen e ndryshoreve të tipit real gjuha programuese Turbo Pascal rezervon 4 ose 8 bit.

Deklarimi i ndryshoreve bëhet në bllokun **VAR** të programit me fjalën e rezervuar REAL.

```
VAR x,y,z: REAL;
```

Me deklarin e ndryshoreve rezervohet memorje për ruajtjen e ndryshoreve të numër real në domenin e dhënë.

Për dallim nga ndryshoret numër i plotë ndryshoret numër dhjetor operatorë DIV dhe MOD nuk implementohen. Ndërs operatorët +,-,* dhe / kanë kuptimin e njejtë me të vetmin dallim se rezultati i njehsuar nga shprehja është numër dhjetor.

Operatori	Përshkrimi
+	Shuma e dy numrave
-	Ndryshimi i dy numrave
*	Prodhi i dy numrave
/	Herësi i dy numrave

Sintaksa e përdorimit të operatorëve aritmetik është:

```
ndryshorja1 operatori ndryshorja2 ose
```

```
shprehja1 operatori shprehja2
```

Shembull

```
x + y
a * b+24
x / 3 +3.4
```

Inicilaizimi i ndryshoreve numër dhjetor mund të bëhet përmes kodit programor përmes urdhërit.

```
Ndryshorja:=shprehja;
```

Ku ndryshorja është e tipit REAL ndërsa shprehja e tipit real ose numër i plotë.

Paraqitja në ekran e ndryshoreve të tipit numër dhjetor bëhet përmes urdhërit WRITE/WRITELN.

```
PROGRAM ParaqitjaNeEkran;
Uses wincrt;
A,B,C: REAL;
BEGIN
  A:=4*5+120;
  WRITELN('Vlera e ndryshores A eshte ', A);
  B:=12.23;
  WRITELN('Vlera e ndryshores B eshte ',B);
  C:=A*B+250;
  WRITELN('Vlera e ndryshores C eshte ',C);
  READKEY;
END.
```

Nëse njehsojmë vlerat e shprehjeve me të cilat inicializohen ndryshoret atëherë vlerat që do të përmbanjnë ndryshoret A, B dhe C janë:

A	140
B	12.23
C	1962.2

Nëse e egzekutojmë programin e dhënë në shembullin e mësipërm në ekran do të paraqitet.

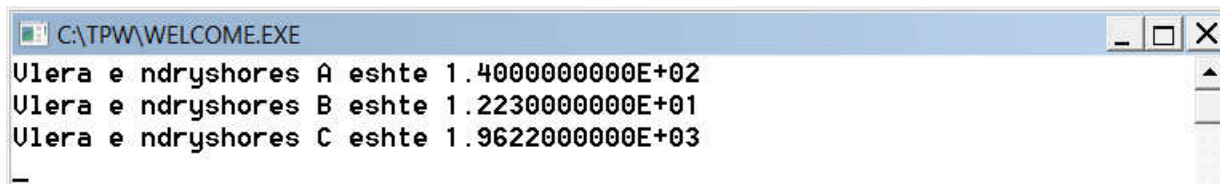


Figura 19 Paraqitja e numrave dhjetor në ekran

Gjuha programuese Turbo Pascal numrat real i paraqet në formën 1.XYZ....E+XP

Ku ndryshet paraqiten si

A	1,40..0*10 ²
B	1.2230..0*10 ¹
C	1.96220..00*10 ³

Për paraqitjen e ndryshoreve real përdoret forma me mantisë dhe eksponent, e cila do të përpunohet në njësinë e ardhëshme mësimore.

Paraqitja në këtë formë e numrave është e vështirë e kupueshme për shfrytëzuesin e programit andaj për ta lehtësuar paraqitjen e numrave përdoret forma:

Writeln(ndryshorja:X:Y);

Ku X dhe Y janë numra të plotë dhe

- X paraqet numrin e hapsirave që rezervohen për paraqitjen e ndryshores
- Y paraqet numrin e shifrave që paraqiten në ekran pas presjes dhjetore

Për vlerat e tipit real, mund ta përdorim sintaksën e lartëpërmendur që në ekran numrat dhjetorë të paraqiten në mënyrë të kuptueshme për shfytëzuesin, gjithashtu bëhet edhe rrumbullaksimi i numrit. Nëse shkruajmë kodin programor në vazhdim.

```
PROGRAM FormatimiNumraDhjetor;
Uses wincrt;
VAR A: REAL;
BEGIN
A:=12.12347;
WRITELN('VLERA E NDRYSHORES A ESHTTE ',A);
WRITELN('VLERA E NDRYSHORES A ESHTTE ',A:8:2);
WRITELN('VLERA E NDRYSHORES A ESHTTE ',A:8:4);
WRITELN('VLERA E NDRYSHORES A ESHTTE ',A:10:4);
READKEY;
END.
```

Në ekran do të parqitet



Figura 20 Formatimi i numrave dhjetor

Inicialiaimi përmes tastierës së ndryshoreve numër dhjetor bëhet përmes urdhërave READ/READLN. Gjatë futjes së vlerës së numrit dhjetor përmes tastierës pjesa e plotë nga pjesa dhjetore ndahet me pikë(.).

Kodi i programit i cili ilustron inicializimin e ndryshoreve numër real përmes tastierës është:

```
PROGRAM Inicializimi;
Uses wincrt;
VAR
    A,B:REAL;
BEGIN
WRITE('SHKRUAJ VLEREN PWR NDRYSHOREN A ');
READLN(A);
WRITE('SHKRUAJ VLEREN PWR NDRYSHOREN B ');
READLN(B);
WRITELN('VLERAT E NDRYSHOREVE JANE');
WRITELN('A=',A:5:2,' B=', B:5:2);
READKEY;
END.
```

Gjatë ekzekutimit të programit vlerat për ndryshoret A dhe B mund të jipen përmes tastierës sikurse në

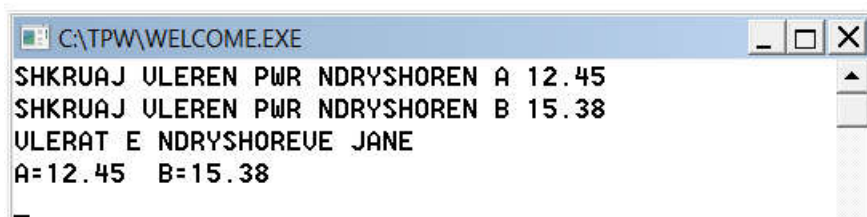


Figura 21 Paraqitja në ekran e numrave dhjetor

Prioritet i operatorëve për numra dhjetor është i njëjtë me prioritetin e operatorëve për numra të plotë. Prioritet më të lartë gjatë njehësimit të shprehjeve kanë operatorët (*) dhe (/) , prioritet më të ulët kanë operatorët (+) dhe (-), Nëse kemi dy operatorë me prioritet të njëjtë atëherë njehësimi i shprehjes fillon nga me majta në të djathtë; ndërsa nëse përdorim kllapa atëherë përparësi ka njehsimi i shprehjes brenda kllapave. **Rregullat për llogaritje të shprehjeve aritmetike janë ekuivalente me rregullat aritmetike.**

2.9.1. FUNKSIONET STANDARDE ME NUMRA DHJETOR

Mënyra dhe qëllimi i përdorimit të funksioneve standarde me numra dhjetor është e njëjtë sikurse me numra të plotë. Tabela e funksioneve për numra të plotë vlen edhe për numrat dhjetor. Me përjashtim të funksioneve RANDOM, PRED, SUCC, ORD, ODD të cilat mund të përdoren vetëm për numra të plotë.

Për funksionet matematike si:

ABS, SIN, SQR, SQRT mund ti referohemi tabelës së funksioneve standarde për numra të plotë me dallimin e vetëm se rezultati kthyes i funksionit është numër dhjetor.

Ndërsa funksionet si në tabelën vijim janë funksione të cilat janë të definuara për parametra numra dhjetor.

Funksioni	Përshkrimi
ROUND(X)	Rrumbullakson numrin X
TRUNC (X)	Pjesa e plotë e numrit dhjetor
FRAC(X)	Pjesa dhjetore e numrit X
INT(X)	Konverton numrin X në numër të plotë

Tabela 6 Funksionet me numra dhjetor

Programi në vijim ilustron përdorimin e funksioneve me numra dhjetor:

```
PROGRAM shembulli;
Uses wincrt;
VAR X,Y:REAL;
BEGIN
X:=55.25;
Y:=55.75;
WRITELN('ROUND(',X:4:2,')=',ROUND(X));
WRITELN('ROUND(',Y:4:2,')=',ROUND(Y));
WRITELN('TRUNC(',X:4:2,')=',TRUNC(X));
WRITELN('FRAC(',X:4:2,')=',FRAC(X):5:2);
WRITELN('INT(',X:4:2,')=',INT(X):5:2);
READKEY;
END.
```

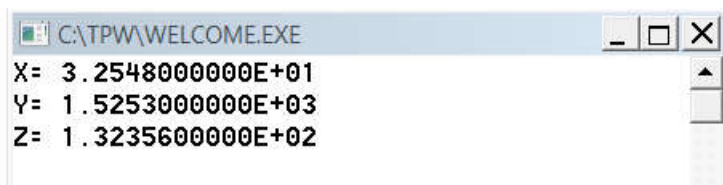
Rezultati i egzekutimit të programit do të jetë:



Figura 22 Funksionet standarde me numra real

Ushtrime:

1. Ndryshoret X Y dhe Z të tipit real janë parqitur në ekran përmes urdhërit writeln të paformatuar. Të gjinden vlerat e tyre nëse vlerat e dhëna janë si në figurë



1. Çka do të paraqesin në ekran urdhërat
 - a. WRITELN(12.456:5:1)
 - b. WRITELN(12.456:1:5)
 - c. WRITELN(1.46:5:3)
2. Të gjendet shprehja ekuivalente aritmetike e shprehjeve dhe të llogaritet vlera e tyre nëse ndryshoret X,Y dhe Z kanë vlerat: X=12.7, y=4.5 dhe Z=1.1
 - a. $X*Y+2*Z$
 - b. $Z*(X+Y)/2$
 - c. $2*X-X/Y/Z;$
3. Të caktohet tipi dhe vlera e shprehjeve nëse ndryshoret X,Y dhe Z kanë vlerat: X=12.7, y=4.5 dhe Z=1.1:
 - a. $SQR(TRUNC(X)-ROUND(Y));$
 - b. $SQRT(FRAC(X)+INT(Y+Z));$
 - c. $ABS(SQRT(TRUNC(Z))-ABS(X));$

2.10. PARAQITJA E NUMRAVE TË PLOTË NË KOMPJUTER

Të gjitha të dhënat instruksionet në memorjen e kompjuterit ruhen në formë binare do të thotë në 0 dhe 1. Të dhënat numerike ruhen dhe organizohen në formë që janë më lehtë për tu shprehur në gjuhën e makinës për ekzekutimin e operacioneve matematikore dhe për ruajtjen e tyre në memorjen e kompjuterit.

0,1 –bit

00110011- bajt (8 bita)

1100 gjysëmbajt (4 bit ang. nibble)

00110011 00110011 fjalë (16 bit ang word)

Paraqitja e numrave të plotë në kompjuter bëhet si në figurën vijuese:

S	0	1	0	.	.	.	1	0
---	---	---	---	---	---	---	---	---

Biti i parë tregon shenjën e numrit nëse $S=0$ numri është pozitiv nëse $S=1$ numri është negativ. Numrat pozitiv paraqiten ashtu që numri dekad konvertohet në binar dhe biti i parë është 0.

Numrat negativ fitohen nga pozitiv duke gjetur komplementin e parë dhe të dytë të numrit të kundërt pozitiv.

Komplementi i parë i një numri binar fitohet ashtu që 0 shëndrrohen në 1-she dhe 1-she konvertohen në 0.

$$00110100_2 \rightarrow 11001011_2$$

Komplementi i dytë fitohet ashtu që komplementi i parë rritet për 1.

$$11001011_2 + 1_2 = 11001100_2$$

$$00110100_2 = 52_{10}$$

$$11001100_2 = -52_{10}$$

n- numri i bitave

shembull:

010101= shifra e parë 0 tregon se numri është pozitiv, ndërsa $10101_2 = 21_{10}$

Shembull për $n=8$;

Numri më i madh pozitiv është $2^{n-1}-1$

$$2^{8-1}-1=2^7-1=127$$

7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	1

Numri më i vogël negativ -2^{n-1}

$$-2^{8-1} = -2^7 = -128$$

7	6	5	4	3	2	1	0
1	0	0	0	0	0	0	0

Zero

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

2.11. PARAQITJA E NUMRAVE DHJETOR ME PIKË LËVIZËSE

Ruajtja e numrave dhjetor në memorjen e kompjuterit sipas standardit me Pikë lëvizëse (ang. floating point), standard i cili rrumbullakson numrat dhjetor duke pasur parasysh precizitetin dhe rangun e tyre. Çdo numër dhjetor mund të paraqitet në formën:

$$X = M * Bazat^E$$

Për numrat dekad vlen $X = M * 10^E$, M-Mantisa E – eksponenti

Numri 23.05 paraqitet si:

$$23.05 = 0.235 \cdot 10^2, \quad M=0.235, \quad E=2$$

$$23.05 = 23500 \cdot 10^{-3}, \quad M=23500, \quad E=-3$$

Sigurisht që mënyra e parë në krahasim me mënyrën e dytë është më e përshtatshme pasi që paraqitet me më pak shifra (0) për paraqitjen e vlerës së njejtë. Edhe për numrin binar mund të shkruhet.

$$11.01 = 0.1101 \cdot 2^{10}, \quad M=0.1101_2$$

$$0.00101 = 0.101 \cdot 2^{-10}$$

$$M=0.101_2$$

$$E=-10_2$$

Mantisa e normalizuar te numrat binar do të thotë pas presjes dhjetore është e pozicionuar 1.

Shembull:

$$10010.0101_2 = 0.100100101 \cdot 2^{101}, \quad E=101_2 = 5_{10}$$

$$0.00010111_2 = 0.101111 \cdot 2^{-11}, \quad E=-11_2 = -3_{10}$$

Tek kompjuterët personal përdoret standardi IEEE, në të cilin mantisa ka vlerën $1 \leq M < 2$ ose në intervali (1.00.... 0 deri në 1.11...1) në formën $1+F$, F është pjesa dhjetore (0.00...0 deri në 0.11...1).

$M=1+F$ njishi i parë gjithmonë nënkuptohen dhe nuk është patjetër të harxhohet bit për shkruarjen e tij. Përdoret një bit plus për të shkruar eksponentin.

Shembull

$$1110.101_2 = 1.110101 \cdot 2^{11}, F=0.110101, E=11$$

Forma e përgjithshme e numrit binar i paraqitur me presje lëvizëse sipas IEEE standardit:

S	$E+2^{e-1}-1$	F
1 bit	e bit(eksponent)	m bit (mantisa decimale)

S shenja e numrit

Eksponenti mund të jetë negativ, dhe me shtimin e $2^{e-1}-1$ (bëhet lëvizje), dhe fitohet $E+2^{e-1}-1$ gjithmonë të jetë numër pozitiv në intervalin 00..0 dhe 11..1 me gjatësi **e** bit.

Sipas IEEE standardit

Me vargun 32 bit paraqiten numrat decimal me precizitet normal (standard).

Për eksponent $e=8$ bit

Mantis deicimale $F=23$ bit

Ndërsa me fjalët 64 bit paraqiten numrat decimal me **precizitet të dyfishtë**.

Për eksponent $e=11$ bit

Mantisë decimale $F=52$;

Ushtrim:

Numri binar 111.011 ($7,375_{10}$) të paraqitet me precizitet standard

Precizitet standard 32 bit.

$$111.011_2 = 1.11011 \cdot 2^{10}, F=11011_2, E=10_2, S=0$$

Lëvizja e eksponentit për 127_{10}

$$2_{10} + 127_{10} = 129_{10} = 10000001_2$$

Në memorjen e kompjuterit numri 111.011 sipas standardit IEEE me 32 bit shprehet si.

01000000100000000000000010000001

2.12. NDRYSHORET E TIPIT KARAKTER

Tipi CHAR (karakter) përdoret për të ruajtur karakter të vetëm. 'A', 'a', '3', '\$', '+' etj. Ndryshorja e tipit karakter deklarohet në bllokun var si:

```
VAR ch: CHAR;
```

Vlerat karakter ruhen në sasi të memorjes prej 8 bitëve (1 bajt). Kodimi universal për kompjuterë personal është ASCII (American Standard Code For Information Interchange), i cili përmban 256 karaktere.

Vlerat të cilat mund të përmbajë ndryshorja e tipit karakter janë:

- të gjitha shkronjat e alfabetit anglez[a..z]ASCII [97..122] [A..Z] ASCII [65..90]
- shifrat [0..9]ASCII[48..59]

Kodi ASCII gjithashtu përmban shenjat e pikësimit dhe karakteret kontrolluese [~, !, #, %, ;, &...].

gjysma tjetër e kodit është e pastandardizuar dhe varet nga makina llogaritëse.

Çdo karakter në ASCII përmban **numrin rendor**, shembull i ASCII kodit është numri rendor për **a** është 97, për **b** është 98, për 0 është 48, për numrin 1 është 49.

Inicializimi përmes kodit programor i ndryshores karakter bëhet ashtu që vlera e karakteri futet në thonjëza të njëfishtë.

```
ch:='A';
```

Nëse vlera që duhet të merr ndryshorja ch është thonjëza e njëfishtë atëherë urdhëri duhet të shkruhet si:

```
ch:=' ';
```

Përveç përmes kodit programues ndryshoret karakter mund të jipet vlerë edhe përmes tastierës me urdheërin READLN;

```
READLN(ch);
```

Gjate inicializimit të ndryshores karakter përmes tastierës nuk është e nevojshme që vlera të futet në thonjëza;

Parqitja në ekran i ndryshoreve karakter bëhet në mënyrën e njëjtë sikurse ndryshoret numër i plotë me urshërin WRITE/WRITELN.

Formatimi për paraqitje më elegante të ndryshoreve të tipit char bëhet në mënyrën e njejtë sikurse me ndryshoret e tipit numër i plotë.

Pas ndryshores vendosen: dhe pastaj shkruhet numri i hapsirave të cilat do të rezervohen për ndryshoren.

Shembull: `writeln('a':3);`

Në ekran do të paraqesë dy hapsira dhe pastaj shkronjën **a**.

		a
--	--	---

Me programin e dhënë do të ilutrojmë inicializimin e ndryshores shenjë përmes kodit programor dhe tastierës dhe paraqitjen dhe formatimin e ndryshores char në ekran.

```
PROGRAM TipiShenje;  
Uses wincrt;  
VAR  
shenja1, shenja2:CHAR;  
BEGIN  
shenja:='A';  
WRITELN('VLERA E NDRYSHORES shenja ESHTË ', shenja:2);  
shenja:='$';  
WRITELN('VLERA E NDRYSHORES shenja ESHTË ', shenja:3);  
WRITE('SHKRUJ VLERË PER NDRYSHOREN: ');  
READLN(shenja);  
WRITE('VLERA E DHENË PER NDRYSHOREN ESHTË ', shenja:4);  
READKEY;  
END.
```

Pas ekzekutimit të programit në ekran do të paraqitet dritarja ku do të shfaqet vera e ndryshoreës shenja dhe në rreshtin e mundësihet që shfrytëzuesi të shkruaj vlerë për ndryshoren shenja(shfrytëzuesi ka zgjedhur (*), dhe e njejta shenjë e formatuar paraqitet në ekran duke rezervuar 4 hapsira dhe rrafshuar nga e djathta.

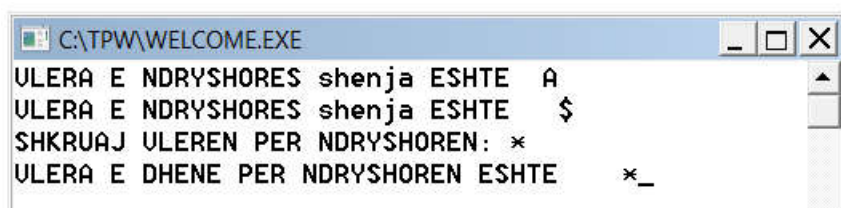


Figure 23 Inicializimi, paraqitja në ekran dhe formatimi i ndryshoreve të tipit shenjë

Disa nga funksionet më të rëndësishme nga libreria e Turbo Pascalit të cilat mundësojnë punë me të dhëna të tipit karakter janë paraqitur në tablën në vazhdim:

Funksioni	Përshkrim
Chr(x)	Funksioni ka parametër numër të plotë [0..255] dhe jep karakterin i cili i përgjigjet numrit rendor nga tabela e ASCII kodit
Ord(ch)	Funksioni ka parametër karakter dhe kthen vlerën numër të plotë [0..255] i cili i përgjigjet karakterit nga tabela e ASCII kodit
Pred (ch)	Funksioni kthen vlerë karakter dhe jep parardhësin e parametrit ch
Succ(ch)	Funksioni kthen vlerë karakter dhe jep pasardhësin e parametrit ch
uppCase(ch)	Funksioni kthen vlerë karakter, shkronjën e madhe të parametrit ch

Shembull nëse kemi të deklaruar ndryshoret **ch** e tipit karakter dhe **x** e tipit numër i plotë segmenti i kodit në vazhdim;

Funksionet standarde do ti ilustrojmë me programin në vazhdim.

```

PROGRAM FunksioneStandarde;
Uses wincrt;
VAR
    shenja:CHAR;
    x:INTEGER;
BEGIN
    x:=99;
    shenja:='c';
    WRITELN('PRED(',shenja,')= ', PRED(shenja));
    WRITELN('SUCC(',shenja,')= ', SUCC(shenja));
    WRITELN('ORD(',shenja,')= ', ORD(shenja));
    WRITELN('CHR(',X,')= ', CHR(x));
    READKEY;
END.

```

kodi i mësipërm në ekran do të paraqesë:

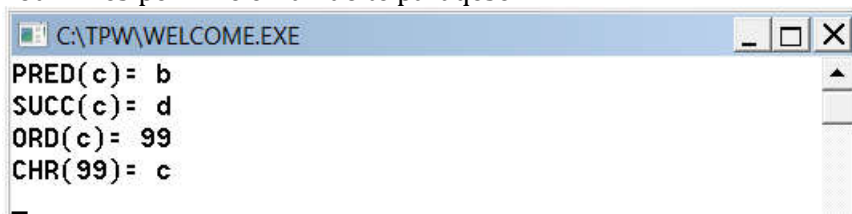


Figure 24

2.13. TË DHËNAT E TIPIT VARG KARAKTERËSH (TEKST)

Në aplikacione të ndryshme kemi të dhëna të cilat duhet të ruhen dhe përpunohen si tekst ose varg karakteresh, të cilat përdoren nga shfrytëzuesit.

Në gjuhën programuese Pascal mund të ruanim të dhëna tekstuale ose varg karakteresh në thonjëz të njefishta si:

- 'Programimi për vitin e dytë, për shkollat e mesme profesionale'
- 'Mësonje programimin me gjuhën programuese Pascal'
- '(a+b)=(b+a)'
- '+38975123456'
- 'Bleron Gashi' etj.

Të dhëna të cilat gjuha programuese Turbo Pascal i njeh si të dhëna të tipit STRING.

Të dhënat e tipit string deklarohen si:

```
VAR s: STRING;
```

Në ndryshoren s mund të ruhen varg karakteresh me gjatësi deri në 255 karaktere. Nëse jemi të sigurtë se vargu me karaktere nuk do të kalojë numër të caktuar karakteresh që është më i vogël se 255 atëherë mund të caktojmë gjatë deklarimit të ndryshores string numrin e karaktereve gjegjësisht gjatësinë e vargut tekstual duke shkruar gjatësinë si:

```
VAR s: STRING[30];
```

Atëherë gjatësia maksimale e vargut tekstual mund të jetë deri në 30 karaktere. Inicializimi i ndryshoreve të tipit string në kodin e programit bëhet nëpërmjet operatorit := (ndryshorja:= shprehja)

```
s:='Programimi për vitin e dytë';
```

kurse përmes tastierës bëhet me urdhërat **read** dhe **readln**.

```
Readln(s);
```

Paraqitje në ekran

```
write('Tung':10,5:4,5673:2);
```

Në ekran do të shkruaj:

1.1. OPERATORËT RELACIONAL

Për të krahasuar vlerat numerike të ndryshoreve përdoren operatorët relacional. Rezultati i krahasimeve të dy ndryshoreve ose shprehjeve numerike është i saktë (TRUE) ose i pasaktë (FALSE);

Operatorët relacional në Paskal janë:

Operatori	Përshkrimi
<	Më e vogël
<=	Më e vogël ose e barabartë
>	Më e madhe
>=	Më e madhe ose e barabartë
=	E barabartë
<>	E ndryshueshme

Sintaksa e përdorimit të operatorëve relacional është:

Shprehja1 OPERATORI RELACIONAL *Shprehja2*

Shprehja 1 dhe *shprehja2* mund të jenë shprehje çfarodo tipi numerike, tekstuale ose karakter.

Operatorët relacional krahasojnë shprehjet dhe vlera të cilën e japin mund të jetë TRUE (e vërtetë) ose FALSE (e pavërtetë) varësisht nga vërtetësia e shprehjes.

Shembull

4<15	TRUE
5*6+2<6*4+1	FALSE
'C'<'D'	TRUE
-4.5>3.5	FALSE

Operatorët relacional dhe formatimi i paraqitjes së ndryshoreve karakter në ekran është i njëjtë me ndryshoret e tipit numër i plotë. Operatorët relacional krahasojnë vlerën e kodit ASCII

'B'>'A' jep vlerën **TRUE** e vërtetë

'e'<'c' jep vlerën FALSE e pavërtetë

'B'>'A' jep vlerën TRUE e vërtetë

'e'<'c' jep vlerën FALSE e pavërtetë

'0'>'A' jep vlerën TRUE e vërtetë

'e'<'c' jep vlerën FALSE e pavërtetë

3. CIKLET DHE DEGËZIMET NË PROGRAME

Qëllimet e temës

- Ti njohë strukturat kontrolluese për përsëritje dhe zgjedhje (degëzim) (**FOR, WHILE , REPEAT – UNTIL**);
- Të njohë urdhërat për përsëritje (**FOR, WHILE , REPEAT – UNTIL**);
- Të njohë urdhërat për degëzim (**IF, CASE**)
- Të njohë përdorim gjegjës të cikleve
- Të njohë përdorimin gjegjës të urdhërave për degëzim
- Të zgjidhë detyra me **FOR** cikël
- Të zgjidhë detyra me ciklin **WHILE**
- Të zgjidhë detyra me ciklin **REPEAT-UNTIL**
- Të zgjidhë detyra me urdhërin **IF**;
- Të zgjidhë detyra me urdhërin **CASE**
- Të zgjidhë detyra të kombinuara me **FOR** ciklet dhe **IF** degëzimet
- Të zgjidhë detyra të kombinuara me **WHILE** ciklet dhe **IF** degëzimet
- Të zgjidhë detyra të kombinuara me **DO WHILE** ciklet dhe **IF** degëzimet

3.1. STRUKTURAT PËR DEGËZIM IF

Programet në kapitullin e mëparshëm egzekutoheshin në radhë të caktuar njëri pas tjetrit duke filluar nga urdhëri i parë i shkruar.

Gjatë procesit të programimit vijmë në situata në të cilat duhet që programit të varësisht nga kushti i dhënë të zgjedhë nga dy ose më tepër opsione që do ti egzekutojë ashtuqë një apo më tepër urdhëra të egzekutojë apo mos të egzekutojë. Varësisht nga kushti i dhënë të zgjedhë sekuençë të urdhërave ose të mos egzekutojë urdhër ose bashkësi urdhërash të dhënë.

Nëse **fjalëkalimi është i saktë atëherë** Mundëso qasje në sistem

Fjalinë do ta ndajmë në këtë mënyrë

1. Nëse **Fjalëkalimi është i saktë**

2. Atëherë **Mundëso qasje në sistem**

Kushti **FJALËKALIMI ËSHTË I SAKTË** mund të ketë vetëm njërën nga vlerat ose **i vërtete** ose **i pavërtetë** është tregues se a do ta kemi qasje në sistem ose jo. Urdhëri për kushtëzim mund të paraqitet edhe grafikisht me bllok diagram.

Sintaksa e urdhërit për kushtëzim në Pascal është

```
IF Kushti  
    THEN
```

```
Urdhëri;
```

Urdhëri për kushtëzim fillon me fjalën kyçe IF, pas saj vendoset shprehje logjike Kushti, shprehje logjike e cila mund të ketë vlerë true (e vërtetë) ose false (e pavërtetë).

Pas kushtit pason fjala kyçe THEN. Dhe në fund shkruhet urdhëri i cili do të egzekutohet atëherë kur kushti ka vlerë TRUE. Nëse shprehja Kushti ka vlerë TRUE atëherë egzekutohet Urdhëri përndryshe urdhëri kalohet pa u egzekutuar. Urdhërin për kushtëzim mund të kuptohet lehtë nga shembulli në vazhdim: Nëse numri i viteve të personit është më i vogël se 18 atëherë në ekran do të paraqitet porosia se nuk lejohet qasja në të dhënat.

```
Program Kushtezimi;  
uses wincrt;  
VAR mosha:byte;  
BEGIN  
    WRITELN('Shkruaj moshen');
```

```
READLN(mosha);
IF mosha < 18
  writeln('Nuk mund te keni qasje nw te dhenat e programit')
  readKey;
END.
```

Shprehja $mosha < 18$ ka mund të ketë vlerë True ose False, teksti Nuk mund të keni qasje në të dhënat e programit paraqitet në ekran varësisht nga vlera e ndryshores mosha, nëse vlera e ndryshores mosha është më e madhe se 18 atëherë do të paraqitet taks i mësipërm në ekran, përndryshe do të ekzekutohet programi në urdhërin në vazhdim.

Nëse në vend të një urdhëri i cili varet nga kushti kemi dy ose më tepër urdhëra atëherë në Pascal. Varësisht nga vlera e shprehjes kushti do të ekzekutohen urdhërat urdhëri_1 deri tek urdhëri_n. Në gjuhën programuese Pascal për të ekzekutuar dy ose më shumë urdhëra që varen nga kushti i dhënë duhet që urdhërat të vendosen në mes të fjalëve kyçe **BEGIN** dhe **END**.

```
IF kushti
  THEN
  BEGIN
    urdhëri 1;
    urdhëri 2;
    .....
    urdhëri n;
  END;
```

Shembull: Të shkruhet programi i cili për vlerat e dhëna të dy ndryshoreve a dhe b nëse $a > b$ atëherë ndryshore shkëmbejnë vlerat.

```
Program VleraAbsolute;
uses wincrt;
VAR a,b,nd:integer;
BEGIN
  WRITELN('Shkruaj vleren për ndryshoren a');
  READLN(a);
  WRITELN('Shkruaj vleren për ndryshoren b');
  READLN(b);

  IF a>b
  THEN
  BEGIN
    nd:=a;
    a:=b;
    b:=nd;
  END;
  WRITELN('a=', a, 'b=', b);
  readKey;
```

END.

Pas egzekutimit të programit të mësipërm nëse vlerat e dhëna par ndryshoret a dhe b plotësojnë kushtin që ndryshorja a të ketë vlerë më të madhe se ndryshorja b ndryshoret do ti shkëmbejnë vlerat, por nëse vlera e ndryshores a nuk është më e madhe se vlera e ndryshores b atëherë vlerat e tyre nuk do të ndryshojnë, ndërsa paraqitja e vlerave ne ekran do të egzekutohet pavarësisht nga kushti.

Detyrë: Si do të ishte rrjedha e programit nëse nuk do të përdornim urdhrat **BEGIN** dhe **END**, Sqaro rastin nëse kushti ka vlerën True dhe nëse kushti ka vlerën False.

Forma të cilën e sqaruam përdorimin e urdhërit për kushtëzim është forma më e thjeshtë e urdhërit për kushtëzim, ofron egzekutimin e një urdhëri nëse shprehja kushti ka vlerë TRUE. Nëse marrim shembullin e mëparshëm me moshën e personit do të thotë nëse shfrytëzuesi ka më pak se 18 vite atëherë ne ekran paraqitet porosia se "Nuk keni qasje në të dhënat e programit" përndryshe paraqitet porosia "Keni qasje në të dhënat e programit".

Urdhëri për kushtëzim mundëson egzekutimin e një nga dy opsionet varësisht nga kushti. Forma e urdhërit për kushtëzim është

```
Nëse Kushti
    atëherë
        Urdhëri_1
përndryshe
        Urdhëri_2
```

Rezultat i egzekutimit do të jetë ose Urdhëri_1 ose Urdhëri_2 varësisht nga shprehja logjike Kushti.

Në Pascal urdhëri për kushtëzim shkruhet në formën:

```
IF Kushti
    THEN
        Urdhëri_1
    ELSE
        Urdhëri_2;
```

Urdhëri për kushtëzim fillon me fjalën kyçe **IF**, pas saj vendoset shprehje logjike Kushti, shprehje logjike e cila mund të ketë vlerë true (e vërtetë) ose false (e pavërtetë).

Pas kushtit pason fjala kyçe **THEN**, pas saj shkruhet urdhëri_1 urdhër i cili egzekutohet nëse kushti ka vlerë TRUE.

Nëse shprehja Kushti ka vlerë FALSE atëherë egzekutohet Urdhëri_2.

Edhe pse në fillim kemi cekur se çdo urdhër mbaron me pikëpresje (;) pas urdhërit Urdhëri_1 nuk vendoset pikëpresje por shkruhet fjala kyçe ELSE dhe pason Urdhëri_2

Shembull: Ndryshorja x e tipit numër i plotë të tregohet nëse është çift ose tek.

```
Program CiftTek;
uses wincrt;
VAR x:integer;
BEGIN
WRITELN('Shkruaj vlerwn per ndryshoren x');
READLN(x);

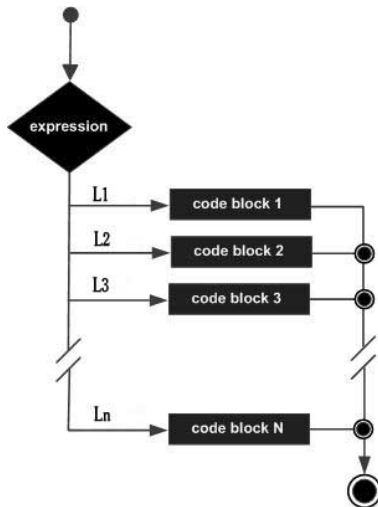
IF x mod 2 = 0
THEN
  WRITELN('numri i dhene ', x, ' eshte numer cift')
ELSE
  WRITELN('numri i dhene ', x, ' eshte numer tek');
readKey;
END.
```

Shprehja $x \bmod 2$ njehson mbetjen gjatë pjesimit të numrit x me 2. Nëse mbetja është 0 atëherë numri është çift dhe kushti ka vlerë TRUE, përndryshe (mbetja është 1) kushti ka vlerë FALSE atëherë numri x është numër tek.

3.2. STRUKTURA PËR DEGËZIM CASE

Struktura IF mundwson zgjidhjen e e problemit pwr degwzim por nw disa raste mund qw kodi tw duket i jo i kuptueshm, prandaj duhet pwrdorur struktura CASE. Case struktura është l ngjajshëm me urdhërin IF, por përjashton shprehjet e tipit string. Sintaksa e përgjithshme për urdhërin case është.

```
case (Shprehja) of
  L1 : S1;
  L2: S2;
  ...
  ...
  Ln: Sn;
end;
```



Ku L1,L2,...Ln janë emra të rasteve , ose vlera hyrëse të cilat mund të jenë shprehje të tipit Integer, karakter ose tip I numërueshëm I të dhënave. S1, S2,... Sn janë urdhëara në Pascal. Shembull.

```

program NotaPerNxenes;
uses wincrt;
var
    nota:Integer;
begin
    Write('shkruaj noten: '); Readln(nota);
    case (nota) of
        5 : writeln('Shkelqyeshem!' );
        4: writeln('Shume mire' );
        3 : writeln('Mire' );
        2 : writeln('Mjaftueshem' );
        1: Writeln('Punoni pak me shume');
        else
            writeln('nuk keni shkruar note ');
    end;
    writeln('Nota juaj eshte ', nota );

    readln;
end.
  
```

Ushtrim: Ndryshorja e tipit karakter e futur perms tastiers te tregohet se cilit tip është. Nese është '0' .. '9' Ne ekran tregohet mesazhi se keni futur numër, 'a'..'z' ose 'A'..'Z' shkronjë , (+,-,*,/) thuhet se është operator.

3.3. STRUKTURAT CIKLIKE

Ciklet janë struktura kontrolluese që mundësojnë që egzekutimi i ndonjë apo më tepër urdhërave të përsëritet.

3.3.1. STRUKTURA WHILE-DO

Kjo komandë përdoret për realizimin e strukturës përsëritje me dalje në fillim. Sintaksa e komandës është.

```
WHILE KUSHTI DO  
    URDHERI ;
```

Domethënia (Semantika) e urdhërit është. Derisa (**While**) kushti plotësohet, egzekuto **Urdherin**. Nëse ka më shumë urdhëra bllok, vendosen në mes të **BEGIN** dhe **END**.

```
WHILE KUSHTI DO  
    BEGIN  
        URDHERI_1 ;  
        URDHERI_2 ;  
        ...  
        URDHERI_N ;  
    END ;
```

Do të thotë fjalët **WHILE** dhe **DO** janë fjalë të rezervuara. Kushti është një shprehje që llogaritet para çdo egzekutimi të komandës ose bllokut të komandave. Vlera e kushtit mund të jetë TRUE ose FALSE. Nëse është True ekzekutohen komandat pas kushtit DO, por nëse është FALSE nuk ekzekutohen por kalohen te komandat tjera pas mbarimit të urdhërave. Kujdes duhet ti kushtohet që kushti të arrijë në mosplotësimin e tij. Që të mos përsëriten urdhërat pakufi.

Shembull:

```
N:=1;  
While N<=0 do  
    BEGIN  
        Write(N:3);  
        N:=N+1;  
    END;
```

Mund të ndodhë që urdhërat e ciklit asnjëherë most ë ekzekutohen.

Ushtrimi 1: Të gjendet shuma e n numrave të parë natyror.

Ushtrimi 2: Të gjendet prodhimi i n numrave të parë natyror.

Ushtrimi 3: Të caktohet numri n a është i thjeshtë ose jo.

Ushtrimi 4: Të thjeshtohet thyesa a/b.

3.3.2. STRUKTURA REPEAT-UNTIL

Kjo komandë përdoret për strukturën përsëritje me dalje në fund ciklit.

Sintaksa e komandës është:

```
Repeat  
    Urdhera;  
Until Kushti;
```

Strukturata REPEAT- UNTIL Përsërit egzekutimin e urdhërave deri në plotësimi i kushtit. Kushti mund të jenë shprehje relacion dhe mund të ketë vlerën e njëjës nga shprehjet logjike TRUE ose FALSE. Pas egzekutimit të urdhërave pas Repeat, ndryshohen variablat e kushtit në cikël, me crast mund të ndryshojë edhe vlera e kushtit në fund të ciklit. Pas çdo ekzekutimi të urdhërave pyetet a është plotësuar kushti, nëse kushti nuk është plotësuar përsëritet egzekutimi i urdhërave, dhe nëse është plotësuar, vazhdohet me urdhërin pas UNTIL. Do të thotë përsëritja mbaron kur kushti do të marrë vlerën **true**. Komandat që ekzekutohen me REPEAT nuk ka nevojë të vendosen në bllok në mes të Begin dhe End.

Shembull të gjendet shuma e numrave prej 1 deri në 5.

```
K:=0;
REPEAT
    K:=K+1;
    S:=K+S;
UNTIL K>5;
```

Ushtrime:

1. Të gjendet shuma e shifrave të një numri të plotë

```
Program ShumaShifrave;
uses wincrt;
VAR shuma,shifra:integer;
    N,nd:longint;

BEGIN
    write('Shkruaj numrin n: ');
    readln(n);
    nd:=n;
    shuma:=0;
    repeat
        shifra:=n mod 10;
        shuma:=shuma+shifra;
        n:=n div 10;
    until n=0;
    writeln('shuma e shifrave e numrit',nd,' eshte ',shuma);

    readln;
    END.
```

2. Të gjendet sa shifror është numri i dhënë n.

3.3.3. STRUKTURA FOR- TO-DO

Kjo komandë përdoret për realizimin e strukturave për përsëritje me numërim të cikleve. Përsëritja numërohet me numërues të veçantë për të cilin shtohet vlerat e fillimit dhe mbarimit të numëruesit. Sintaksa e komandës është

```
FOR ndryshorja := VleraFillestare TO VleraPerfundimtare DO  
    Urdheri;
```

Domethenia e urdherit është Për vlerat e ndryshores nga një vlerë e fillimit, duke u rritur për 1 deri në një vlerë përfundimtare, të ekzekutohet **Urdheri**.

Shembulli 1.

```
For i:=1 TO 5 DO  
    Write('* ');
```

Shembulli 2:

```
FOR shkronja:='A' to 'Z' DO  
    Writeln(shkronja);
```

-Ndryshoret (I dhe shkronja) paraqesin numëries të cikleve dhe kjo duhet të jetë e tipit linear të rregulluar, dhe nuk duhet të jetë ndryshore e tipit real.

-Vlera e fillimit dhe vlera përfundimtare e ndryshores mund të jetë konstantë ose shprehje, por duhet të jetë e tipit të njejtë.

-Këto vlera nuk duhet të ndryshohen pas fjalës DO .

- Ndryshores automatikisht I caktohet vlera e ardhshme per ciklin e ardhshëm. Në të cilin ekzekutohet urdhëri

-Nëse ka më shumë urdhëra pas DO, atëherë ato vendosen në bllok mes BEGIN dhe END.

3.3.4. STRUKTURA FOR-DOWNTO-DO

Sintaksa e komandës është:

```
FOR ndryshorja:= vlera fillestare DOWNTO vleraPerfundimtare DO  
    Urdheri;
```

Për vlerat nga një vlerë fillestare duke e zvogëluar ate nga 1 deri në një vlerë përfundimtare të ekzekutohen Urdhëri në vijim.

Shembull:

```
FOR I:=15 DOWNTO 5 DO  
    WRITELN(I);
```

Ushtrime:

1. Shkruaj programin I cili kerkon që të shkruhen n numra të plotë dhe gjen maksimumin.
2. Te shkruhet program I cili gjen numrin prodhimin e numrave tek deri te numri n.
3. Te shkruhet programi I cili cakton shumen $\sum_{i=1}^n \frac{x-i}{3}$ (i nga 1 deri ne n), ku x dhe n jane variabla hyrese.

Referencat

1. Turbo Pascal, Reference Manual, ALPHA SYSTEM CORPORATION
2. Essential Pascal, Marco Cantù
3. Turbo Pascal Programmer's Guide 6.0, Borland International
4. Turbo Pascal Programmer's Reference 7.0, Borland International
5. Learn Pascal, Sam A. Abolrous
6. <http://www.tutorialspoint.com/pascal/>
7. www.wikipedia.org
8. Programimi pwr vitin e IV, Julijana Petreska
9. <http://www.freepascal.org/>
10. Informatika edubuntu , Tekst për vitin I të arsimit të mesëm profesional- për të gjitha profesionet.