**Zad 1.**

SUSJEDNI(X[]:FLOAT,N:INTEGER)
I,P:INTEGER
MN:FLOAT
P=1
MN=X[1]-X[2]
I=2
WHILE I≤N-1
IF X[I]-X[I+1]<MN
P=I
MN=X[I]-X[I+1]
ENDIF
I=I+1
ENDWHILE
OUTPUT X[P],X[P+1]

**Zad 2.**

CIFRE(S[]:CHAR,X[]:INTEGER,N:INTEGER):INTEGER
I,J,OS,IND:INTEGER
I=1
WHILE S[I]≠’\0’
IF S[I]≥’0’ ∧ S[I]≤’9’
OS=CHAR2INT(S[I])-CHAR2INT(‘0’)
IND=0
J=1
WHILE J≤N
IF X[J]≡OS
IND=1
ENDIF
J=J+1
ENDWHILE
**IF IND≡0
RETURN 0
ENDIF**ENDIF
I=I+1
ENDWHILE
RETURN 1

**Zad 3.**

SUMALR(X[]:INTEGER,N:INTEGER):INTEGER
I,SL,SR,S:INTEGER
S=0
FOR I=1:N
S=S+X[I]
ENDFOR
SL=0
SR=S-X[1]
FOR I=2:N
IF SL≡SR
RETURN I-1
ENDIF
SL=SL+X[I-1]
SR=SR-X[I]
ENDFOR
IF SL≡SR
RETURN N
ELSE
RETURN-1
ENDIF

4) Potprogram, niz cijelih (prirodni) brojeva, zamjeniti broj sa cifrom desetica tog broja, jednocifrene brojeve zamjeniti sa -1.

JEDINICE(&X[]:INTEGER,N:INTEGER)
I,Q:INTEGER
FOR I=1:N
IF X[I]≤9
X[I]=-1
ELSE
Q=X[I]/10
X[I]=Q-(Q/10)\*10
ENDIF
ENDFOR

5) Dat je sortirani niz i broj T. Odrediti bilo koji par u nizu takav da je X[J]/X[I]=T (X[J]=T\*X[I]), T>1

T=3 3->12

1 3 7 12 15 28 O(N2)
BINAR(X[]:INTEGER,L:INTEGER,R:INTEGER,B:INTEGER):INTEGER
M:INTEGER
WHILE R≥L
M=(R+L)/2
IF X[M]≡B
RETURN M
ELSEIF X[M]<B
L=M+1
ELSE
R=M-1
ENDIF
ENDWHILE
RETURN -1
----------------------

N,I,A[1000],T,IND:INTEGER
INPUT N,T
FOR I=1:N
INPUT A[I]
ENDFOR
IND=0

I=1
WHILE I≤N-1∧IND≡0

IF BINAR(A,I+1,N,T\*A[I])>0

IND=1
OUTPUT A[I],T\*A[I]
ENDIF
I=I+1
ENDWHILE
IF IND≡0
OUTPUT “NEMA TOGA”
ENDIF
END

6) Data je struktura koje se zove NEPAR. Niza cijelih brojeva i njegove dužine. Napisati f-ju koja prima dvije promjenljive tipa NEPAR.F-ja vraća 1 ako su svi niza prve strukturne promjenljive neparni, vraća 2 ako prethodno nije zadovoljeno i ako su svi elementi niza druge strukturne promjenljive neparni. U suprotnom f-ja treba da vrati -1.

TYPE NEPAR
X[1000]:INTEGER
N:INTEGER
ENDTYPE

FJA(S1:NEPAR,S2:NEPAR):INTEGER
I,M1,M2:INTEGER
M1=0
M2=0
FOR I=1:S1.N
IF S1.X[I]-(S1.X[I]/2)\*2≡1
M1=M1+1
ENDIF
ENDFOR

FOR I=1:S2.N
IF S2.X[I]-(S2.X[I]/2)\*2≡1
M2=M2+1
ENDIF
ENDFOR
IF M1≡S1.N
RETURN 1
ELSEIF M2≡S2.N
RETURN 2
ELSE
RETURN -1
ENDIF

8) Struktura VELIKO, koja se sastoji od dva cijela broja X i Y i stringa B sa makar jednim velikim slovom u sebi. Funkcija prima prima dva podatak tipa VELIKO, vraća kao rezultat prvo veliko slovo u onoj strukturi koja ima veće X.
TYPE VELIKO
X,Y:INTEGER
B[1000]:CHAR
ENDTYPE

FX(V1:VELIKO,V2:VELIKO):CHAR
I:INTEGER
IF V1.X>V2.X
I=0
WHILE V1.B[I+1]≠’\0’
I=I+1
IF V1.B[I]≥’A’∧V1.B[I]≤’Z’
RETURN V1.B[I]
ENDIF
ENDWHILE
ELSE
I=0
WHILE V2.B[I+1]≠’\0’
I=I+1
IF V2.B[I]≥’A’∧V2.B[I]≤’Z’
RETURN V2.B[I]
ENDIF
ENDWHILE
ENDIF

9) Struktura STUDENT: imena studenta, broja pol ispita i niza ocjena na ispitima.
TYPE STUDENT
IME[1000]:CHAR
BPI:INTEGER
OC[100]:INTEGER
ENDTYPE

SREDNJA\_OCJENA(S:STUDENT):FLOAT
I,T:INTEGER
T=0
FOR I=1:S.BPI
T=T+S.OC[I]
ENDFOR
RETURN IN2FLOAT(T)/S.BPI

I,TI,J,N,SO:INTEGER
S[130]:STUDENT
PO[130],SPO:FLOAT
INPUT N
TI=0
SO=0
FOR I=1:N
INPUT S[I].IME,S[I].BPI
TI=TI+S[I].BPI
FOR J=1:S[I].BPI
INPUT S[I].OC[J]
SO=SO+S[I].OC[J]
ENDFOR
PO[I]=SREDNJA\_OCJENA(S[I])
ENDFOR
FOR I=1:N
OUTPUT S[I].IME,S[I].BPI,PO[I]
ENDFOR
OUTPUT INT2FLOAT(SO)/TI
END

10) Struktura KOLOKVIJUM, SALA -> String, niz cijeli brojeva BROJ\_INDEKSA, niz cijeli brojeva GODINA\_UPISA, BROJ\_STUDENATA (cijeli broj)

TYPE KOLOKVIJUM
SALA[50]:CHAR
BROJ\_STUDENATA,BROJ\_INDEKSA[150]:INTEGER
GODINA\_UPISA[150]:INTEGER
ENDTYPE

K:KOLOKVIJUM
I,NAJ:INTEGER
NAJ=2021
INPUT K.SALA,K.BROJ\_STUDENATA
FOR I=1:K.BROJ\_STUDENATA
INPUT K.BROJ\_INDEKSA[I],K.GODINA\_UPISA[I]
IF K.GODINA\_UPISA[I]<NAJ
NAJ=K.GODINA\_UPISA[I]
ENDIF
ENDFOR
FOR I=1:K.BROJ\_STUDENATA
IF K.GODINA\_UPISA[I]≡NAJ
OUTPUT K.BROJ\_INDEKSA[I],K.GODINA\_UPISA[I]
ENDIF
END

7) 

TYPE LISTA
S[100]:CHAR
\*NEXT[]:LISTA
ENDTYPE

LEXY(&GL[]:LISTA):INTEGER
IF GL.NEXT≡0
RETURN 1
ELSEIF POREDISTR(GL.S,GL.NEXT.S)>0
RETURN 0
ELSE
RETURN LEXY(GL.NEXT)
ENDIF

11) Klasa Automobil, podaci članovi
Marka->string, starost ->cio broj, kilometri->cio broj, potrošnju na 100km -> realan broj.

Barem: 2xkonstruktor, 2xmutator, 2xinspektor

Najveci

CLASS AUTOMOBIL
MARKA[1000]:CHAR
STAROST:INTEGER
KILOMETRI:INTEGER
POTROSNJA:FLOAT
AUTOMOBIL()
AUTOMOBIL(S[]:CHAR,X:INTEGER,Y:INTEGER,Z:FLOAT)
GET\_ST():INTEGER
GET\_KM():INTEGER
GET\_POT():FLOAT
SET\_ST(S:INTEGER)
SET\_KM(S:INTEGER)
SET\_POT(S:FLOAT)
FRIEND NAJVECI(A[]:AUTOMOBIL,K:INTEGER):AUTOMOBIL
ENDCLASS

AUTOMOBIL() FROM AUTOMOBIL

AUTOMOBIL(S[]:CHAR,X:INTEGER,Y:INTEGER,Z:FLOAT) FROM AUTOMBIL
STRCOPY(S,MARKA)
STAROST=X
KILOMETRI=Y
POTROSNJA=Z

GET\_POT():FLOAT FROM AUTOMOBIL
RETURN POTROSNJA

SET\_KM(S:INTEGER) FROM AUTOMOBIL
KILOMETRI=S

NAJVECI(A[]:AUTOMOBIL,K:INTEGER):AUTOMOBIL
I:INTEGER
B:AUTOMOBIL
B=A[1]

FOR I=2:K
IF A[I].STAROST>B.STAROST ∨(A[I].STAROST≡B.STAROST ∧ A[I].KILOMETRI>B.KILOMETRI) ∨(A[I].STAROST≡B.STAROST ∧ A[I].KILOMETRI≡B.KILOMETRI∧A[I].POTROSNJA>B.POTROSNJA)
B=A[I]
ENDIF
ENDFOR

RETURN B

13) Klasa RADNIK. Podaci članovi: Ime string, Broj – identifikacioni broj (cijeli broj), Koeficijent – realni broj. Konstruktori, inspektori, mutatori

CLASS RADNIK
IME[100]:CHAR
BROJ:INTEGER
KOEF:FLOAT
RADNIK()
RANDIK(A:INTEGER)
SET\_BROJ(A:INTEGER)
SET\_KOEF(A:FLOAT)
GET\_BROJ():INTEGER
GET\_KOEF():FLOAT
STAMP(R:RADNIK)
ENDCLASS

STAMPAJ(R:RADNIK) FROM RADNIK
IF R.KOEF<KOEF
OUTPUT IME
ELSE
OUTPUT R.IME
ENDIF

14) TACKA, X, Y koordinate tačke (realni brojevi)

Rastojanje između dvije tačke (funkcija članica)
klasa KRUG podatak član CENTAR, R poluprečnik kruga->povrsina I obim kruga

CLASS TACKA
X,Y:FLOAT
DIST(T:TACKA):FLOAT
ENDCLASS

CLASS KRUG
CENTAR:TACKA
R:FLOAR
OBIM():FLOAT
POVRS():FLOAT
ENDCLASS

DIST(T:TACKA):FLOAT FROM TACKA
RETURN **SQRT**((X-T.X)\*(X-T.X)+(Y-T.Y)\*(Y-T.Y))

OBIM():FLOAT FROM KRUG
RETURN 2\*R\*3.1415926

POVRS():FLOAT FROM KRUG
RETURN R\*R\*3.1415926

12) Preformulisan!!!

Pacijent: Ime, BN (broj nalaza), niz NALAZA (niz realnih brojeva)

MIN f-ja clanica koja prima kao argument cijeli broj

MIN(1)

CLASS PACIJENT
IME[100]:CHAR
BN:INTEGER
NALAZI[50]:FLOAT
MIN(K:INTEGER):FLOAT
SORT()
ENDCLASS

MIN(K:INTEGER):FLOAT FROM PACIJENT
**THIS.SORT()**RETURN NALAZI[K]

SORT() FROM PACIJENT
I,J:INTEGER
T:FLOAT
FOR I=1:BN-1
FOR J=I+1:BN
IF NALAZI[I]>NALAZI[J]
T=NALAZI[I]
NALAZI[I]=NALAZI[J]
NALAZI[J]=T
ENDIF
ENDFOR
ENDFOR