

- ۱- برنامه ای بنویسید که معدل ۱۰۰ دانشجو را گرفته و در نهایت فقط معدلهای بالاتر از میانگین کل کلاس را زیر هم نمایش دهد.
- ۲- تابعی بنویسید که ورودی آن یک آرایه ۵ در ۶ بوده و خروجی آن تعداد عناصر منفی آن ماتریس باشد.
- ۳- برنامه ای بنویسید که نام و معدل ۳۰ دانشجو را گرفته و سپس اسامی دانشجویان را به ترتیب معدل، زیر هم چاپ کند.
- ۴- رکورد `Material` برای توصیف یک ماده شیمیایی را در نظر بگیرید:

```
Type Material = Record
  name: String;      { نام ماده شیمیایی }
  m, v: Real         { جرم و حجم ماده شیمیایی }
end;
```

- تابعی (به نام `Density`) برای محاسبه چگالی هر ماده شیمیایی توصیف شده با رکورد فوق، بنویسید.
- ۵- تابعی بنویسید که که ورودی آن آرایه ای ۵۰ عضوی از `Material` بوده و خروجی آن حاصلجمع کل جرمهای مواد داده شده باشد.
  - ۶- برنامه ای بنویسید که اطلاعات ۵۰ ماده شیمیایی (`Material`) را گرفته و در نهایت نام ماده ای که دارای کمترین چگالی است را چاپ کند.
  - ۷- زیربرنامه ای برای ترسیم یک ماتریس ۵ در ۴ داده شده - به شکل منظم - روی مونیتر، بنویسید.
  - ۸- برنامه ای بنویسید که ۱۰۰ عدد را گرفته و اولین عدد تکراری وارد شده را مشخص و چاپ کند.
  - ۹- تابعی بنویسید که به ازای آرایه ۱۰۰ عضوی داده شده (متشکل از اعداد صحیح)، اگر آن آرایه به شکل صعودی مرتب است نتیجه آن `True` و در غیر اینصورت `False` باشد. (نام تابع را `ascending` بگذارید)
  - ۱۰- ساختاری برای توصیف کالای یک فروشگاه شامل: نام کالا، قیمت خرید، قیمت فروش و وزن کالا تعریف کنید. از اسمهای زیر برای نامیدن استفاده کنید:

• نام ساختار = article

• قیمت خرید = pbuy

• قیمت فروش = psale

• وزن = weight

۱۱- با داشتن تعریف ساختار article تابعی به نام interest بنویسید که به ازای یک کالای داده شده، درصد سود حاصل از فروش آن را برگرداند. راهنمایی:

$$\text{قیمت خرید} \div (\text{قیمت خرید} - \text{قیمت فروش}) \times 100 = \text{درصد سود}$$

۱۲- با داشتن ساختار article و تابع interest برنامه ای بنویسید که اطلاعات ۵۰۰ کالای فروخته شده یک فروشگاه را گرفته و موارد زیر را چاپ کند:

(a) مجموع وزن همه کالاها

(b) نام گرانقیمت ترین کالا از نظر قیمت فروش

(c) تعداد کالاهایی که دارای سودی بیش از ۲۵ درصد بوده اند.

۱۳- برنامه ای بنویسید که حاصل عبارت زیر را به دست آورده و نمایش دهد:

$$S = \frac{1+2}{1 \times 2} + \frac{2+3}{2 \times 3} + \frac{3+4}{3 \times 4} + \dots + \frac{99+100}{99 \times 100}$$

۱۴- برنامه ای بنویسید که دو عدد صحیح را گرفته و تعداد مقسوم علیه های مشترکشان را بنویسد.

۱۵- برنامه ای بنویسید که تعداد نامشخصی عدد صحیح را از کاربر بگیرد و با وارد شدن اولین عدد منفی گرفتن اعداد را قطع کرده و میانگین اعداد مثبت فرد وارد شده را نمایش دهد.

۱۶- برنامه ای بنویسید که عناصر یک ماتریس ۴ در ۵ را گرفته و

(a) مشخص کند چندتای آنها صفر است

(b) آن ماتریس را به شکل منظم روی مونیتر چاپ کند.

۱۷- رکوردی به نام fbclub برای توصیف یک باشگاه فوتبال با فیلدها و اسامی پیشنهادی زیر تعریف کنید:

• نام باشگاه (name)

• تعداد برد (won)

- تعداد مساوی (equ)
- تعداد باخت (loss)
- تفاضل گل (gavg)

۱۸- زیربرنامه ای به نام `getfc` برای دریافت اطلاعات یک رکورد `fbclub` بنویسید.

۱۹- تابعی به نام `fcscores` برای محاسبه امتیاز باشگاهی که به عنوان پارامتر ورودی می گیرد، بنویسید. (هر برد معادل ۳ امتیاز و مساوی معادل ۱ امتیاز است)

۲۰- با فرض داشتن موارد فوق، تابعی به نام `fccompare` برای مقایسه برتری امتیازی دو باشگاه فوتبال بنویسید. توجه کنید در امتیاز مساوی، تفاضل گل بیشتر مبنای برتری است.

۲۱- با فرض داشتن همه موارد قبل، برنامه ای بنویسید که اطلاعات ۱۸ باشگاه فوتبال را بگیرد و در نهایت نام باشگاه قهرمان را نمایش دهد.

----- سوالات جدید: -----

۲۲- رکوردی به نام `bazikon` برای توصیف یک بازیکن بنویسید که شامل نام بازیکن (`nam`) و سن (`sen`) وی باشد.

۲۳- رکوردی بنام (`team`) بنویسید که شامل نام (`nam`) تیم و آرایه ای یازده عضوی از بازیکنان باشد.

۲۴- زیربرنامه ای به نام `getba` بنویسید که وظیفه آن دریافت اطلاعات یک بازیکن از صفحه کلید باشد.

۲۵- زیربرنامه ای به نام `getteam` بنویسید که وظیفه آن دریافت اطلاعات یک تیم از صفحه کلید باشد.

۲۶- تابعی بنام `msen` بنویسید که ورودی آن یک تیم و خروجیش میانگین سنی بازیکنان آن تیم باشد.

۲۷- به کمک جواب تمرینهای ۲۲ تا ۲۶ برنامه ای بنویسید که اطلاعات ۲۰ تیم را گرفته و اسامی تیمهایی را که میانگین سنی زیر ۲۵ سال دارند زیر هم چاپ کند.

پاسخها از صفحه بعد ←

```
{ ----- 1 ----- }
var x: array[1..100] of real;
    avg, sum:real;
    i: integer;
begin
    for i:=1 to 100 do
        begin
            read(x[i]);
            sum := sum+x[i];
        end;
    avg:=sum/100;
    for i:=1 to 100 do
        if x[i]>avg then
            writeln(x[i])
        end;
    end.
```

```
{ ----- 2 ----- }
type matrix = array[1..5, 1..6] of real;
function neg(x: matrix):integer;
var n, i, j:integer;
n:=0;
for i:=0 to 5 do
    for j:=0 to 6 do
        if x[i, j]<0 then
            n:=n+1;
        end;
    end;
neg:=n
end;
```

```
{ ----- 3 ----- }
var name: array[1..30] of string;
    avg: array[1..30] of real;
    tn: string;
    ta: real;
    i, j: integer;
begin
    for i:=1 to 30 do
        begin
            readln(name[i]);
            readln(avg[i])
        end;
    end;
```

```
end;

for i:=29 downto 1 do
  for j:=1 to i do
    if avg[j]<avg[j+1] then
      begin
        ta:=avg[j];
        avg[j]:=avg[j+1];
        avg[j+1]:=ta;
        tn:=name[j];
        name[j]:=name[j+1];
        name[j+1]:=tn
      end;
    end;
  end;

  for i:=1 to 30 do
    writeln(name[i])
  end.

{ ----- 4 ----- }
type material=record
  name:string;
  m, v:real
end;

function density(x: material):real;
begin
  density:= x.m/x.v
end;

{ ----- 5 ----- }
type arrmat=array[1..50] of material;

function summ(var x: arrmat):real;
var s:real;
    i: integer;
begin
  s:=0;
  for i:=1 to 50 do
    s:=s+x[i].m;
```

```
        summ:=s
end;

{ ----- 6 ----- }
var a: arrmat;
    i: integer;
    minden:real;
    minname:string;
begin
    minden:=1e38;
    for i:=1 to 50 do
        with a[i] do
            begin
                readln(name);
                readln(m, v);
                if density(a[i])<minden then
                    begin
                        minden:= density(a[i]);
                        minname:=name
                    end
                end
            end;
        end;
    end;

    writeln(minname)
end.

{ ----- 7 ----- }
type matrix=array[1..5, 1..4] of real;

procedure tarsim(x:matrix);
var i, j: integer;
begin
    for i:=1 to 5 do
        begin
            writeln;
            for j:=1 to 4 do
                writeln(x[i,j]:6:2)
            end
        end
    end;
end;
```

```
{ ----- 8 ----- }
var x:array[1..100] of integer;
  i, j: integer;
begin
  for i:=1 to 100 do
    read(x[i]);

  for i:=1 to 100 do
  begin
    for j:=1 to i-1 do
      if x[j]=x[i] then
        break;

    if x[j]=x[i] then
    begin
      writeln(x[i]);
      break
    end
  end
end.
end.
```

```
{ ----- 9 ----- }
type arr=array[1..100] of integer;

function ascending(x: arr): boolean;
var i: integer;
  t:boolean;
begin
  t:=true;
  for i:=2 to 100 do
    if x[i]<x[i-1] then
    begin
      t:=false;
      break
    end;

  ascending:=t
end;
```

```
{ ----- 10 ----- }
type article=record
  name: string;
  pbuy, psale:integer;
  weight: real
end;

{ ----- 11 ----- }
function interest(x:article): real;
begin
  interest:= (x.psale-x.pbuy)/x.pbuy*100
end;

{ ----- 12 ----- }
var a: array[1..50] of article;
  i, ig, nsud: integer;
  sumw: real;
begin
  ig:=1;
  for i:=1 to 50 do
  with a[i] do
  begin
    readln(name);
    readln(pbuy, psale, weight);
    sumw:=sumw+weight;
    if psale> a[ig].psale then
      ig:=i;

    if interest(a[i])>25 then
      nsud:=nsud+1
  end;

  writeln(sumw); { a }
  writeln(a[ig].name); { b }
  writeln(nsud) { c }
end.

{ ----- 13 ----- }
var i:integer;
```

```
s:real;
begin
  for i:=1 to 99 do
    s:= s+ (i + i+1)/(i*(i+1));

  writeln(s)
end.
```

```
{ ----- 14 ----- }
var x, y, i, n: integer;
begin
  read(x, y);
  for i:=1 to x do
    if (x mod i=0) and (y mod i = 0) then
      n:=n+1;
  writeln(n)
end.
```

```
{ ----- 15 ----- }
var x, s, n: integer;
begin
  while true do
    begin
      read(x);
      if x<0 then
        break;

      if x mod 2 = 1 then
        begin
          s:=s+x;
          n:=n+1
        end
    end;

  writeln(s/n)
end.
```

```
{ ----- 16 ----- }
var x: array[1..4, 1..5] of real;
```

```
nsefr, i, j: integer;
begin
  for i:=1 to 4 do
    for j:=1 to 5 do
      begin
        read(x[i, j]);
        if x[i, j]=0 then
          nsefr:=nsefr+1
        end;
      end;
    end;
  writeln(nsefr); { a }

  for i:=1 to 4 do { b }
  begin
    writeln;
    for j:=1 to 5 do
      write(x[i, j]:6:2)
    end
  end.

{ ----- 17 ----- }
type fbclub=record
  name:string;
  won, loss, equ, gavg: integer
end;

{ ----- 18 ----- }
procedure getfc(var x:fbclub);
begin
  with x do
  begin
    readln(name);
    readln(won, loss, gavg)
  end
end;

{ ----- 19 ----- }
function fcscor(x:fbclub):integer;
begin
```

```
    fcscorres:=x.won*3+x.equ  
end;
```

```
{ ----- 20 ----- }  
function fccompare(a, b: fbclub):boolean;  
begin  
    if fcscorres(a) = fcscorres(b) then  
        fccompare := a.gavg>b.gavg  
    else  
        fccompare:= fcscorres(a) > fcscorres(b)  
    end;  
end;
```

```
{ ----- 21 ----- }  
var a: array[1..18] of fbclub;  
    i, ichamp: integer;  
begin  
    ichamp:=1;  
    for i:=1 to 18 do  
        begin  
            getfc(a[i]);  
            if fccompare(a[i], a[ichamp]) then  
                ichamp:=i  
            end;  
            writeln(a[ichamp].name)  
        end.  
end.
```

```
{ ----- 22 ----- }  
type bazikon = Record  
    nam: string;  
    sen: integer  
end;
```

```
{ ----- 23 ----- }  
type team = record  
    nam: string;  
    b: array[1..11] of bazikon  
end;
```

```
{ ----- 24 ----- }
```

```
procedure getba(var x: bazikon);
begin
    readln(x.nam);
    readln(x.sen)
end;
```

```
{ ----- 25 ----- }
procedure getteam(var x: team);
var i: integer;
begin
    readln(x.nam);
    for i:=1 to 11 do
        getba(x.b[i])
end;
```

```
{ ----- 26 ----- }
function msen(x: team):real;
var i: integer;
    s: real;
begin
    s:=0;
    for i:=1 to 11 do
        s:=s+x.b[i].sen;
    msen := s/11
end;
```

```
{ ----- 27 ----- }
var a: array[1..20] of team;
    i:integer;
begin
    for i:=1 to 20 do
        getteam(a[i]);
    for i:=1 to 20 do
        if msen(a[i]) < 25 then
            writeln(a[i].nam)
end.
```