

# نمونه مسائل حل شده بر نامه نویسی پاسکال

با نظارت و تنظیم: گنجی

# مثالهای معمولی

\*\*\* علامتهای ستاره میزان سختی سوال را نشان می دهند.

۱) \* برنامه ای را بنویسید که اعداد فرد سه رقمی را از بزرگ به کوچک چاپ کند.

```
var n, i, x : integer;
begin
  writeln('this is a tuorun program');
  i:=999;
  while i>100 do
  begin
    writeln(i);
    i:=i-2
  end
end.
```

تذکره:

این برنامه یک برنامه خود کار است یعنی به محض اجرا خود کار خواسته شده را تحویل می دهد.

۲. \*\* برنامه ای بنویسید که دو عدد صحیح را بگیرد و اعداد زوج بین آنها را چاپ کند.

در واقع کلماتی مثل بگیرد یا دریافت کند نشان دهنده باز بودن مساله است یعنی اینکه حداقل باید کامپایلر چیزی را یا از طریق صفحه کلید یا هر چیز ورودی دیگر که بوسیله کاربر وارد می شود را بخواند یا دریافت کند.

```
var a,b,x,y :integer;
begin
  writeln('Enter number One =');
  readln(a);
  writeln('Enter number Two =');
  readln(b);
  if a < b then
  begin
    x:=a;      { قسمت اول }
    y:=b;
  End
  Else
  begin
    x:=b;
    y:=a;
  end;
  if x mod 2 <>0 then      { قسمت دوم }
    x:=x-1;
  while x<y do
  begin
    x:=x+2;
    Writeln(x)
  end
end.
```

در این برنامه ابتدا ما با استفاده از یک دستور `if` بزرگی و کوچکی هر یک از اعداد را چک نموده ایم بدین صورت که فرض بر آن است که کاربر اعداد را به ترتیب وارد نکند یعنی مثلاً کاربری ابتدا عدد بزرگتر و سپس کوچکتر را تایپ کند و کاربری دیگر برعکس. لذا این برنامه با استفاده از قسمت یک عدد کوچکتر را در ظرف `x` و عدد بزرگتر را در ظرف `y` قرار می دهد. سپس با استفاده از یک حلقه `while` تعریف کرده ایم که تا زمانی که  $x < y$  است کارهای قسمت دو را به شرح ذیل انجام دهد:

اگر `x` بر `2` بخش پذیر باشد (شرط برقرار باشد) یعنی  $x = x + 2$  هر بار `2` تا به مقدار `x` می افزاید ( تا `y` ، والا یک واحد از `x` کم کن تا زوج شود و به همین شکل ادامه بده.

---

۳) \* برنامه ای بنویسید که ابتدا تعداد دانش آموزان و سپس نمرات آنها را دریافت کند و در نهایت میانگین کل کلاس را چاپ کند.

```
Var  n,i:integer;
      sum, avg, nomreh:real;
begin
  writeln('Enter n students=');
  read(n);
  writeln('Enter num students:');
  for i:=1 to n do
  begin
    read(nomreh);
    sum:=sum+nomreh;
  end;
  avg:=sum/n;
  writeln('teadad:',n);
  writeln('moadel kol class:',avg:6:2)
end.
```

---

۴- \*\* برنامه ای بنویسید که صد عدد را دریافت کند و در پایان کوچکترین و بزرگترین آنها را چاپ کند.

برای حل این برنامه کافی است شما ظرفی مجازی برای اولین عدد وارد شده در نظر بگیرید و نیز دو ظرف دیگر یکی برای بزرگترین عدد و دیگری کوچکترین عدد در نظر بگیرید. سپس دیگر اعداد که از طریق یک حلقه `for` که دارای دو شرط یکی شرط بزرگتر بودن و دیگری کوچکتر بودن است را مورد ارزیابی نسبت به عدد اولی که کاربر وارد کرده می کند بدین صورت که در شرط اول پس از وارد نمودن هر عدد آن را نسبت به عدد اول مقایسه می کند اگر بزرگتر بود آن را در محل ظرف اول می ریزد و به همین صورت تا اینکه بزرگترین عدد در ظرف `m` قرار می گیرد. ولی برای شرط دوم موجود در حلقه عکس این عمل تکرار می شود یعنی کافی است شما اعداد را نسبت عدد اول از لحاظ کوچکی مورد مقایسه قرار دهید.

```
Var  min,x,min:integer;
begin
  max:=-32767;
  min:=32767;
  for i:=1 to 100 do
  begin
    read(x);

    if max < x then
```

```

        max:=x;
    if min > x then
        min:=x;
    end;
writeln('max is:',max);
writeln('min is:',min);
end.

```

نکته:

در این برنامه برای عدد اول دو ظرف تعریف شده یکی max و دیگری min است.

۵- \*\* برنامه ای بنویسید که دو عدد را بگیرد و (ب م م) و (ک م م) آنها را چاپ کند.

```

var  a,b, i, kmm, bmm:integer;
begin
    writeln('Enter two numbers');
    read(a);
    read(b);
    for i:=a downto 1 do
        if (a mod i =0) and (b mod i =0) then
            begin
                bmm:=i;
                break;
            end;
        kmm=a*b div bmm;
        Writeln('BMM=', bmm, ' KMM=', kmm)
    End.

```

برای محاسبه ک.م.م. از تعریف ریاضی آن (ک م م = حاصلضرب دو عدد تقسیم بر ب م م) استفاده شده است.

۶- \*\* برنامه ای بنویسید که جدول ضرب ۱۰\*۱۰ را چاپ کند.

```

Var  i,j:integer;
begin
    for i:=1 to 10 do
        begin
            writeln;
            for j:=1 to 10 do
                write(i*j:8)
            end
        end
    end.

```

نکته ای که حائز اهمیت است اینست که ابتدا حلقه داخلی به ازای هر  $i$  ،  $n=j$  یعنی ۱۰ بار اجرا می شود و بعد مقداری به اضافه می کند و دوباره ...

۷- \*\* برنامه ای بنویسید که حاصل عبارت زیر را حساب کند:

$$1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots$$

حل: علامت جمله مورد محاسبه را (که در واقع صورت کسر آن نیز هست) در متغیر *alamat* قرار داده ایم که با رسیدن به هر جمله جدید قرینه می شود.

```
var i,j, n, alamat:integer;
    sum:real;
begin
    writeln('Enter number ');
    read(n);
    alamat:=1;

    for i:=1 to n-1 do
    begin
        sum:=sum+alamat/i;
        Alama t:= -alamat;
    End;

    writeln('This sum is:',sum:5:2)
end.
```

۸- \*\* برنامه ای بنویسید که حاصل عبارت زیر را حساب کنید:

$$\frac{۴*۶*۸*...*۱۰۰}{۲*۳*۵*۷*...*۹۹}$$

```
Var i:integer;
    hasel:real;
begin
    hasel=1.0; { عضو خنثی ضرب }
    i:=2;
    while i<=100 do
    begin
        hasel:=hasel * i/(i-1);
        i:=i+2
    end;
    writeln(hasel:6:2);
end.
```

۹- \* برنامه ای بنویسید که عددی را بگیرد و جزء صحیحش را چاپ کند.

```
var i :integer;
    x:real;
begin
    writeln('enter number x:');
    read(x);
    while i<=x do
        i:=i+1;
```

```
write('[x] is:', i);  
end.
```

۱۰ - \*\* برنامه بنویسید که عدد صحیحی را بگیرد و با کمترین تعداد دفعات تکرار اول بودن آن را مشخص کند.

توضیح: عدد اول عددی است که اولین مقسوم علیه بزرگتر از یکش از خود عدد کوچکتر باشد. ضمناً بررسی مقسوم علیه‌ها تا جذر عدد مورد نظر کافی است (شرط حلقه را ببینید).

```
Var i,n:integer;  
begin  
  read(n);  
  i:=2;  
  while (i*i<=n) and (n mod i <>0) do { پیدا کردن اولین مقسوم علیه: }  
    i:=i+1;  
  
    if i<n then  
      writeln('add shoma aval nist.');    else  
      writeln('add shoma aval ast.')end.
```

۱۱ - \*\* برنامه‌ای بنویسید که تمام اعداد دو رقمی که بر مجموع ارقامش بخشپذیرند را چاپ کند.

برای انجام این کار باید کارهای زیر در یک حلقه‌ای که از اولین عدد دو رقمی شروع و به آخرین رقم دو رقمی ختم می‌شود را انجام دهد.  
- رقم دهگان (d) = تقسیم صحیح عدد بر ۱۰  
- رقم یکان (y) = باقیمانده عدد بر ۱۰  
- سپس عدد را بر مجموع مقدار ۱ و ۲ تقسیم می‌کند اگر باقیمانده صفر شد یعنی عدد بر مجموع ارقامش بخشپذیر است و آن را چاپ کند وگرنه حلقه بار دیگر با اضافه کردن یکی بر مقدار آ‌کار خود را تا آخرین عددش ادامه می‌دهد.

```
Var y,d,i:integer;  
begin  
  for i:=10 to 99 do  
    begin  
      d:=i div 10;  
      y:=i mod 10;  
      if i mod (y+d) = 0 then  
        writeln(i)  
    end  
end.
```

۱۲ - \* یک عدد حقیقی و یک عدد صحیح مثبت را بگیرد و اولی را به توان دومی برساند

یعنی:  $Hasel := X^n$

```

Var   i,n:integer;
      x,hasel:real;
begin
  hasel:=1.0;
  writeln('Enter one num haghigy:');
  read(x);
  writeln('Enter one num sahih:');
  read(n);
  for i:=1 to n do
    hasel:=hasel*x;

  writeln('X^n:',hasel:8:2)
end.

```

-----

-----

۱۳- \*\*\* برنامه ای بنویسید که عددی را بگیرد و آن را به عاملهای اولش تجزیه کند.

حل: کافی است با یافتن هر مقسوم علیه آنقدر عدد مورد نظر را بر آن تقسیم کنیم تا دیگر بر آن بخشپذیر نباشد.

```

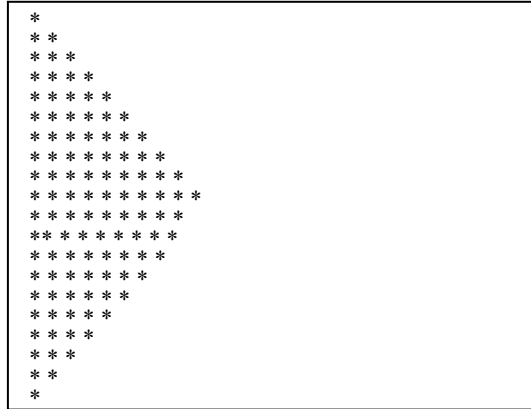
Var   x, i:integer;
Begin
  Writeln('Enter x');
  Read(x);
  i:=2;
  repeat
    if x mod i =0 then
      begin
        writeln(i);
        while x mod i =0 do
          x:=x div i;
        end;
        i:=i+1
      until i>x
  end.

```

-----

-----

۱۴- \*\* برنامه ای بنویسید که روی مانیتور شکلهای زیر چاپ شود. (باکاراکتر \*)  
 بدلیل اینکه برنامه بهتر قابل فهم گردد هر شکل در برنامه ای جدا نوشته شده است.  
 الف:



این قسمت از برنامه کار خاصی ندارد فقط با کمی آشنایی از صفحه مانیتور که چند سطر و چند ستون دارد  
براحتی با استفاده از دو متغیر **آ** و **ا** می توان شکل الف را بر روی مانیتور چاپ کرد.

اگر بخواهیم فاصله بین ستاره های چاپ شده بیشتر باشد می توان بجای دستور `write(*)` از `write(*, ' ')` استفاده کرد که هر چندر بین کوتیشن ها فاصله خالی بیشتری بگذاریم ستاره ها با فاصله بیشتری از هم چاپ می شوند.

برای دیدن شکل می توانید در پایان دستورات قبل از `end` آخر از یک دستور `read` استفاده کنید.  
در برنامه از دو حلقه که هر دو حلقه تودرتو می باشند استفاده شده که حلقه تودرتوی اول کار چاپ نصف شکل را می کند یعنی در واقع هر بار ستاره های به تعداد ستاره های سطر بعد می کند ولی حلقه تودرتوی دوم عکس این کار را انجام می دهد.

**Program test14A;**

**var i,j: integer;**

**begin**

**for i:=1 to 13 do**

**begin**

**for j:=1 to i do**

**write('\*');**

**writeln;**

**end;**

**for i:=12 to 1 do**

**begin**

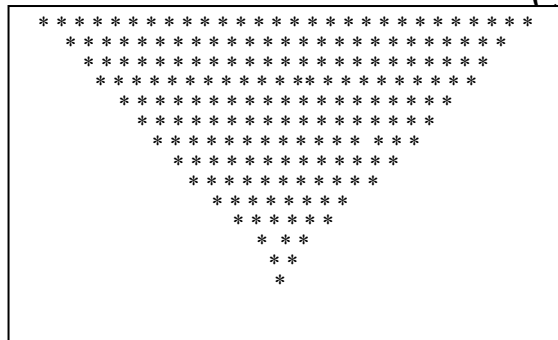
**for j:=12 downto i do**

**write('\*');**

**writeln;**

**end;**

**end.**



**var a,b,c:integer;**

**begin**

```

c:=25;
for a:=1 to 25 do
begin
for b:=1 to 25 do
begin
if (b<c-a) and (b>=a)then
write('*')
else
write(' ');
end;
writeln;
end;
readln
end.

```

به تریس برنامه توجه فرمایید:

a	b	c	b<c-a	b>=a
1	1	25	1<25-1	1>=1
.	2		2<25-1	2>=1
.	3		3<25-1	3>=1
.	.	.	.	.
2	1		1<25-2	1>=2
2	2		2<25-2	2>=2
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

در اینجا وقتی شرط برقرار نیست  
else شرط برقرار است لذا بجای ستاره  
کراکتر خالی می گذارد.

۱۵ - \*\*\* برنامه ای بنویسید که به روش تنصیف معادله زیر را تا چهار رقم اعشار حل

$$x + e^x = 5 \text{ کند.}$$

حل: هدف بدست آوردن ریشه معادله زیر است:

$$F(x) = x + e^x - 5 = 0$$

دو تخمین اولیه مورد نیاز است، یکی عددی که به ازای آن مقدار معادله مثبت باشد ( $Xp$ ) و دیگری که مقدار منفی به معادله بدهد ( $Xn$ ). تخمین جدید در هر مرحله ( $Xm$ ) میانگین دو تخمین قبل است (در کلاس حل تمرین مختصری شرح داده شد).

```

Var      Xp, Xn, Xm:real;
Begin
  Xn=0;      {      F(0) < 0      }
  Xp=3;      {      F(3) > 0      }
  Repeat
    Xm:=(xp+xn)/2;
    If Xm+Exp(Xm)-5 > 0 then {      F(Xm) > 0 یعنی      }
      Xp:=Xm
    Else
      Xn:=Xm
  Until (Xp-Xn)<1E-5;
  Writeln('answer is ', Xm)
End.

```

# مثالهای آرایه ها

۱۶- \* یک لیست (آرایه یک بعدی) از اعداد و سپس یک عدد را بگیرد و تعداد دفعات تکرار آن عدد را در لیست بگیرد.

```
var num:array[1..100] of integer;  
    i,n,s:integer;  
begin  
    writeln('enter 100 numbers:');  
    for i:=1 to 100 do  
        read(num[i]);  
    writeln('Enter num to search:');  
    read(n);  
    for i:=1 to 100 do  
        if n=num[i] then  
            s:=s+1;  
    writeln('tedad tekrar:',s)  
end.
```

این برنامه با برای یک آرایه صدتایی نوشته شده است و برای تعداد اعداد بیشتر می توان هر کجا مقدار ۱۰۰ در برنامه تایپ شده را عدد مورد نظر خود را بگذارید.

در این برنامه از یک حلقه for به طوری استفاده شده است که پس وارد شدن اعداد توسط کاربر عملیات جستجو را بر روی عددی که کاربر جهت جستجو وارد می کند را بر روی کلیه اعداد بررسی می کند بدین صورت که هرگاه عددی از آرایه ها برابر عددی که کاربر داده است بود ظرف s یکی به آن اضافه می شود. و در نهایت پس از پایان حلقه مقدار s را اعلام می دارد. در این جستجو حتی اگر کاربر عددی وارد کند که در آرایه ها نباشد درست عمل کرده و مقدار آن را صفر اعلام می دارد.

-----  
-----  
۱۷- \*\* یک لیست از اعداد را بگیرد و به ترتیب صعودی مرتب کند.

```
var num:array[1..10] of integer;  
    i,j, temp:integer;  
begin  
    writeln('enter 10 numbers:');  
  
    for i:=1 to 10 do  
        read(num[i]);  
  
    for i:=10 downto 1 do  
        for j:=1 to i-1 do  
            if num[j] > num[j+1] then  
                begin  
                    temp:=num[j];  
                    num[j]:=num[j+1];  
                    num[j+1]:=temp;  
                end;  
  
    writeln('adad beh tatib soaoudi is:');  
    for i:=1 to 10 do  
        write(num[i]:3);
```

end.

در این برنامه از روش حبابی که در زیر توضیح می دهیم استفاده شده است .  
روش حبابی: (bubble sort)

در این روش دو عنصر متوالی با یکدیگر مقایسه می شوند، چنانچه اولی از دومی بزرگتر باشد، جای آنها در آرایه عوض می شود. وقتی یک بار تمام عناصر، دو به دو باهم مقایسه شدند، بزرگترین عنصر به انتهای آرایه می رود. دفعه بعد که عناصر آرایه دو به دو باهم مقایسه می شوند، دومین عدد از نظر بزرگی قبل از بزرگترین عنصر، در انتهای آرایه می رود. این روند تا مرتب سازی نهایی ادامه می یابد.

مثال:

9 6 3 4  
6 3 4 9  
3 4 6 9

عناصر اولیه:

مرحله اول:

مرحله دوم:

---

۱۸- \*\* برنامه ای بنویسید که اعداد اول کوچکتر از ۱۰۰۰ را در آرایه ای به ترتیب ذخیره کند.

شرح برنامه:

برای نوشتن این برنامه از یک متغیر بولی بنام w استفاده می کنیم که مشخص می کند عدد مورد نظر مرکب است یا نه. کامپیوتر ابتدا مقدار c را برابر ۱ قرار داده سپس از ۱ تا ۱۰۰۰ عملیات زیر را تکرار می کند :  
مقدار w غلط است اگر برای  $b = (a \div 2)$  به پایین مقداری باشد که باقیمانده a بر b صفر شود و مقدار w صحیح است اگر شرط برقرار نباشد .  
باز در همین حلقه اگر w غلط باشد آنگاه یعنی عدد اول است لذا  $a = q[c]$  در نتیجه در پایان حلقه مقدارهای ذخیره شده در هر آرایه q[c] چاپ می شود.  
اما اگر w صحیح باشد آنگاه حلقه به دور بعد می رود.  
( از این روش برای پیدا کردن اعداد اول در مسائل معمولی شماره ۱۰ نیز می توان استفاده کرد که با کمترین تعداد دفعات تکرار می توان اعداد اول عدد درخواستی را چاپ کرد.)  
نکته:

دقت کنید که وقتی w غلط است حلقه تکراری (حلقه تو در تو)، اجرا می شود تا زمانی که حلقه داخلی پایان پذیرد. یعنی این حلقه خاص زمانی است که w صحیح نمی باشد.

```
var a,b,c:integer;  
    q:array[1..1000] of integer;  
    w:boolean;  
begin  
    C:=1;  
    for a:=1 to 1000 do  
    begin  
        w:=false;  
        for b:=(a div 2) downto 2 do  
        begin  
            if a mod b=0 then  
                w:=true  
            end;  
            if w=false then  
                begin  
                    q [c]:=a;  
                    c:=c+1  
                end;  
            end;  
        end;  
    end;  
    for a:=1 to c do
```

```
write(q [a]:8);  
end.
```

-----  
-----  
۱۹- \*\* برنامه ای بنویسید که لیستی را بگیرد، بزرگترین، مقدار، کوچکترین مقدار، متوسط لیست و واریانس آنرا حساب کند.

```
var x:array[1..100] of real;  
    avg,vari,sum,max,min:real;  
    i:integer;  
begin  
    writeln('enter one numbers of all data');  
    read(x[1]);  
    x[1]:=max;  
    x[1]:=min;  
    writeln('enter remainder data');  
  
    for i:=2 to 100 do  
    begin  
        read(x[i]);  
        sum:=sum+x[i];  
        if max < x[i] then  
            max:=x[i];  
        if min > x[i] then  
            min:=x[i];  
    end;  
    avg:=sum/100;  
  
    for i:=2 to 10 do  
        vari:=vari+((x[i]-avg)*(x[i]-avg))/100;  
    writeln('min is: ',min);  
    writeln('max is: ',max);  
    writeln('sum is: ',sum);  
    writeln('avg is: ',avg);  
    writeln('vari is: ',vari)  
end.
```

در این برنامه ابتدا از کاربر می خواهیم اولین عدد بانک داده های خود را وارد کند و بعد ما آن را سه ظرف ذخیره می کنیم و سپس الباقی اعداد را از او یکی یکی می گیریم . اینکه چرا آن را در دو ظرف به نام min و max وارد می کنیم به این دلیل است که در حلقه for دو مقایسه نسبت به اعداد انجام شود یعنی در یکی کوچکترین عدد و در دیگری بزرگترین عدد ذخیره می شود البته در همین حلقه for کار جمع اعداد را نیز انجام می شود و چون حلقه for از دومین عدد شروع می شود پس مستلزم این بود که از قبل مقدار اولین درایه را در ظرف مجموع بریزم (ظرف سوم) . خب حال برای بدست آوردن میانگین اعداد کافیست مجموع را بر ۱۰۰ تقسیم کنیم و اما برای بدست آوردن واریانس اعداد مان باز مستلزم این است از قبل واریانس اولین عدد را محاسبه کرده باشیم لذا در اینجا به این دلیل که اولین عدد در واقع جزء درایه های آرایه ۱۰۰ عضوی نمی باشد . یعنی در اصل ۹۹ درایه از ۱۰۰ درایه پر می شود. پس حلقه دیگری تعریف می کنیم که فرمول واریانس را برای تک تک اعداد محاسبه کند و آن را هر بار با اجرای حلقه با مقدار قبلی جمع می کند.

-----  
-----

۲۰- \*\* برنامه ای بنویسید که لیستی از اعداد را بگیرد و سپس اعداد مثبت آن را و اعداد منفی آن را جداگانه چاپ کند.

```

Var x:array[1..100] of real;
i: integer;
begin
  writeln('Enter all data');
  for i:=1 to 100 do
    read(x[i]);

    writeln('*****Adad Mosbat*****');
    for i:=1 to 10 do
      begin
        if x[i] >= 0 then
          writeln(x[i]:7:2);
        end;
      writeln(' *****Adad Manfi*****');
      for i:=1 to 100 do
        begin
          if x[i] < 0 then
            writeln(x[i]:7:2);
          end;
        end.

```

این برنامه بنظر می رسد که می شد تنها با دو حلقه for آن را نوشت حدس شما درست است اما آنگاه مشکل تنظیم اعداد مثبت و منفی به صورت دو لیست جداگانه بود. این برنامه ابتدا با استفاده از حلقه for تمامی اعداد را در یک آرایه ۱۰۰ تایی خوانده و ذخیره می کند. سپس با حلقه دوم اعداد مثبت داده شده توسط کاربر را با یک شرط کوچک از کل اعداد جدا کرده و سپس لیست اعداد منفی را چاپ می کند. تذکر: عدد صفر جزء اعداد مثبت فرض شده است.

۲۱- \*\* برنامه ای بنویسید که یک ماتریس ۳\*۳ را بگیرد و دترمینان آن را محاسبه کند.

برای محاسبه دترمینان یک ماتریس ۳\*۳ یک آرایه ۹ تایی برای ذخیره مقدار نه عضو ماتریس کافیست. سپس فرمول دترمینان را به صورت زبان برنامه نوشته و از برنامه می خواهیم مقدار عبارت ریاضی که محاسبه دترمینان ماتریس داده شده توسط کاربر است را بر ایمان چاپ کند.

$$\begin{pmatrix} X_1 & X_2 & X_3 \\ X_4 & X_5 & X_6 \\ X_7 & X_8 & X_9 \end{pmatrix} \Rightarrow \text{دترمینان} = X_1 \begin{pmatrix} X_5 & X_6 \\ X_8 & X_9 \end{pmatrix} - X_2 \begin{pmatrix} X_4 & X_6 \\ X_7 & X_9 \end{pmatrix} + X_3 \begin{pmatrix} X_4 & X_5 \\ X_7 & X_8 \end{pmatrix}$$

```

var x:array [1..9] of real;
i, j:integer;

```

```

det,m,s:real;
begin
for i:=1 to 9 do
  read(x[i]);
  det:=(x[1]*abs((x[5]*x[9])-(x[6]*x[8]))
-(x[2]*abs((x[4]*x[9])-(x[6]*x[7])))+(x[3]*abs((x[4]*x[8])-(x[5]*x[7])));
  writeln('determinan matris 3*3 is: ',det:3:2);
  readln(i)
end.

```

۲۲ - \*\* برنامه ای بنویسید که یک ماتریس ۳\*۳ را بگیرد، مجموع عناصرش را حساب کند و چاپ کند.

```

Var  x:array[1..9] of integer;
      i,j,sum:integer;
begin
  for i:=1 to 3 do
    for j:=1 to 3 do
      begin
        read(x[i,j]);
        sum:=sum+(x[i,j]);
      end;
    writeln('sum matris 3*3 is: ',sum)
  end.

```

برنامه نوشته شده را می توانستیم با استفاده از یک حلقه `for i:=1 to 9` بنویسیم اما برای درک بهتر سطر و ستون های ماتریس آن را به صورت دو حلقه `i` و `j` نوشته ایم. این برنامه را تریس می کنیم: مانیتور

```

9
1
2
6
8
5
7
3
4
sum matris 3*3 is: 48

```

اعداد ۹ و ۱ و ۲ سطر اول و  
۶، ۸ و ۵ سطر دوم و...

x	i	j	sum
0	0	0	0
9	1	1	9
1	1	2	10
.	.	.	.
.	.	.	.
3	3	2	44
4	3	3	48

۲۳ - \*\* برنامه ای بنویسید که دو لیست از اعداد را بگیرد (هر کدام ۱۰۰ عضو) و تعداد اعضای مشترکشان را چاپ کند.

```

var  x:array[1..100] of integer;
      y:array[1..100] of integer;
      i,j,s:integer;
begin
  writeln('type numbers list one');

```

```

for i:=1 to 100 do
    read(x[i]);
writeln('type numbers list two');
for j:=1 to 100 do
    read(y[j]);
    for j:=1 to 100 do
        for i:=1 to 100 do
            begin
                if x[i] = y[j] then
                    s:=s+1;
            end;
        writeln('tedad azay moshterak is: ',s);
    end.

```

این برنامه هر چند در مجموع ۲۰۰ آرایه را می گیرد اما حلقه (3)for برای هر آرایه x، ۱۰۰ بار تکرار می شود یعنی ۱۰۰۰ بار در مجموع. برنامه با استفاده از حلقه های تودر تو و یک شرط براحتی انجام پذیر است. کافی است با دو حلقه for جداگانه ابتدا مقادیر لیست اول و سپس دوم را در دو آرایه ذخیره کنیم سپس به بررسی تک تک آنها بپردازیم که این عمل به این صورت انجام می شود که ابتدا x[i] (مقدار اولین عدد از لیست اول i:=1) با ۱۰۰ درایه از لیست دوم مقایسه می شود که آیا مقدار آن برابر با درایه های y[i] است یا خیر به محض برقرار بودن شرط یک عدد به ظرف کنتور ما اضافه میشود و این روند تا i:=100 ادامه می یابد.

۲۴- \*\* برنامه ای بنویسید که مثلث خیام را تا ۱۰ مرحله چاپ کند.

```

Var khayam:array[1..10, 1..10] of integer;
i, j: integer;
Begin
{ محاسبه }
For j:=1 to 10 do
Begin
    khayam[j, 1]:=1;
    khayam[j,j]:=1;
    For i:=1 to j-2 do
        Khayam[j, i+1]:=khayam[j-1, i]+khayam[j-1, i+1];
    End;
}
{ چاپ }
For j:=1 to 10 do
Begin
    writeln;
    for i:=1 to j do
        write(khayam[j, i]:5)
    End
End.

```

## مثالهای متفرقه

۲۵- برنامه ای که مقدار  $x$  را بر حسب رادیان ( $x$  مثبت است) از ورودی خوانده، سینوس آن را از بسط محاسبه کند (برای ۱۰ جمله):

$$\{ \sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \}$$

از رابطه هر جمله با جمله قبل استفاده می کنیم:

$$\text{جمله قبلی} * (-x^2 / (n * (n-1))) = \text{جمله فعلی}$$

```
Var      Hasel, x, t: real;      { Hasel: Will Be Sin(x) }
          n: integer;          { t: جمله مورد محاسبه }
Begin
  Read(x);
  Hasel:=x;
  n:=3;
  repeat
    t:= -x*x/(n*(n-1))*t;
    Hasel := Hasel + t;
    n := n+2
  Until abs(t)<1E-6;

  Writeln('sin(x)=', Hasel)
End.
```

۲۶- برنامه ای که عناصر ماتریس  $3 \times 3$  را از ورودی خوانده، ترانزاده آن را پیدا می کند و هر دو ماتریس را به خروجی می برد. ترانزاده ماتریس، ماتریسی است که سطرهای آن، ستونهای ماتریس اولیه اند.

```
Var A, B: Array[1..3, 1..3] of integer;
    i, j: integer;
Begin
  { Reading A and Calculating B..... }
  For i:=1 to 3 do
    For j:=1 to 3 do
      Begin
        Read(A [i, j]);
        B[j, i]:=A[i, j]
      End;
  End;
```

```

{ Printing B on the Screen.... }
  For i:=1 to 3 do
    Begin
      Writeln;
      For j:=1 to 3 do
        Write(B [ i, j ]:6:2 );
    End;

```

---

۲۷- \*\* برنامه ای به زبان پاسکال بنویسید که دو عدد n و r را بخواند و مقدار  $C_n^r$  را محاسبه و چاپ کند:

$$C_n^r = \frac{n!}{r! (n - r)!}$$

```

Var      hasel, n, r, surat, makh, i:integer ;
Begin
  Surat:=1;
  Makh:=1;
  For i:=n downto (r+1) do
    Surat:=surat*i;
  For i:=n-r downto 1 do
    Makh:=makh*i;
  Hesel:=surat div makh;
  Writeln('Hasel = ', hasel)
End.

```

---

۲۸- \*\* برنامه ای به زبان پاسکال بنویسید که دو عدد صحیح را از ورودی بخواند و عدد اول را به مبنای عدد دوم تبدیل کند آنگاه نتیجه را با پیغام مناسب در خروجی چاپ نماید.

```

Var m,n,k,r,j:integer ;
Begin
  Write ( 'Enter A Number And Base : ' );
  Read (n,m) ;
  J:=1 ; k:=0 ;

```

```

Write (n, ' in base ', m) ;
Repeat
  R:=n mod m ; n:=m div m ;
  K: =k+r*j ;
  J:=j*10 ;
  Until n<m ;
  K:=n*j+k ;
Until false
WriteLn (' = ', k) ;
End.

```

نمونه ای از خروجی حاصل از اجرای این برنامه به صورت زیر است:

```

Enter A Number And Base : 521 4
521 in base 4 = 20021

```

۲۹- برنامه ای به زبان پاسکال بنویسید که عدد N را از ورودی بخواند و مجموع سری  $1^1+2^2+\dots+N^N$  را محاسبه کرده، نتیجه را با پیغام مناسب در خروجی چاپ کند.

```

Var    l,j,s,a,b,n: Longint ;
Begin
  S :=0 ;
  Write (Enter n : ' ) ;
  ReadLn (n) ;
  For l:=1 to n do
    Begin
      A:=l ;
      B:=1;
      For j:=1 to l do
        B:=a*b ;
        S:=s+b ;
      End ;
    End ;
  Write ( ' Sum Of 1 ^ 1+2^2+3^3+...+', n, ' ^ ', n, ' = ', s) ;
End .

```

نمونه ای از خروجی حاصل از اجرای این برنامه به صورت زیر است:

```

Enter n : 8
Sum Of 1 ^ 1+2^2+3^3+...+8 ^ 8 = 17650828

```