

## Pascal

زبان پاسکال یکی از زبانهای برنامه سازی است که ویژگیهای آموزشی آن بسیار بالاست یعنی زبان خوبی برای آموزش برنامه نویسی منظم و دقیق است. قابلیت خوانایی و درک برنامه های این زبان بالاست و میتوان برنامه هایی نوشت که احتمال خطاهای آنها بسیار اندک باشد علتش این است که یک زبان ساختار یافته (structured) است .

ساختار کلی برنامه چیست؟

۱-تعریف

۲-به وسیله procedure و function قسمتهای تکراری را از بین ببریم

۳-بدنه

فرم کلی پاسکال:

```

Program      اسم ;
const       تعریف ثابت ;
type        تعریف نوع ;
var         تعریف متغیرها ;

procedure   نام      (      پارامترها      )
begin
    _____
    _____
    _____
end ;

function   نوع تابع (      پارامترها      ) نام
begin
    _____
    _____
    _____
end ;
begin
    _____
    _____
    _____
end .
  
```

متغیر : محلی است که دارای یک اسم می باشد و می توان مقداری را در آن ذخیره کرد

طول(بایت)	مقادیری که می پذیرند	نوع
۱	۰ تا ۲۵۵	byte
۱	۱۲۸- تا ۱۲۷	Shortint
۲	۳۲۷۶۸- تا ۳۲۷۶۷	integer
۲	۰ تا ۶۵۵۳۵	Word
۴	۲۱۴۷۴۸۳۶۴۸- تا ۲۱۴۷۴۸۳۶۴۷	Longint

انواع متغیرها در پاسکال:

Integer: انواع عددی صحیح شامل اعداد بدون نقطه ممیز است

انواع داده های صحیح

اعداد اعشاری:

انواع عددی اعشاری شامل اعدادی با نقطه ممیز است

طول(بایت)	ارقام با معنی	مقادیری که میپذیرند	نوع
4	7 تا 8	1.5e-45 تا 3.4e38	single
6	11 تا 12	1.7e38 تا 2.9e-39	real
8	15 تا 16	1.7e308 تا 5.0e-324	Double
10	19 تا 20	1.1e493 تا 1.9e-4951	extended

انواع داده های اعشاری

- Char:

نوع داده کاراکتری که با کلمه char مشخص می شود میتواند یک کاراکتر را در خود نگهداری کند بنابراین کاراکترها در یک بایت از حافظه کامپیوتر ذخیره می شوند

### - String :

این نوع داده ها برای ذخیره رشته ها مثل اسامی افراد به کار می رود و با کلمه کلیدی string مشخص می گردد برای هر رشته باید طول معلوم کنیم اگر طولی تعریف نشود به طور پیش فرض ۸۰ در نظر گرفته میشود (مقدار طول حداکثر ۲۵۵ است)

### - Boolean :

نوع دیگری از داده ها در پاسکال وجود دارند که داده های بولی خوانده می شوند مقادیر بولی دارای دو ارزش درستی یا نادرستی اند که در پاسکال با صفر و یک نمایش داده می شوند صفر به معنی نادرستی و یک به معنی درستی به کار میرود این نوع داده ها با کلمه Boolean تعریف میشوند.

### دستورات ورودی در پاسکال:

دستور Readln برای خواندن اطلاعات از ورودی به کار میرود.  
فرمت آن به صورت مقابل است:

read/ln (اسامی متغیرها)

اسامی متغیرها باید با کاما از یکدیگر جدا شوند..

برای چاپ اطلاعات در خروجی از دستور write/ln استفاده میکنیم.  
فرمت آن به صورت مقابل است:

write/ln( ' یک پیام '  
( متغیر )  
عبارت محاسباتی

### انتساب متغیرها:

مقدار ثابت  
( متغیر ) := متغیر  
عبارت محاسباتی

مثال : برنامه ای بنویسید که دو عدد را از ورودی دریافت و حاصل جمع آن دو را چاپ کند .

```
Program add;
Var
  a,b:integer;
begin
writeln('enter 2 num');
readln(a,b);
```

```
c:=a+b;
writeln('result is ',c);
end.
```

**تکلیف :** چرا برنامه باید ادب داشته باشد ؟

**مثال :** برنامه ای بنویسید که حقوق پایه و تعداد فرزندان یک کارگر را از ورودی دریافت و مبلغ حقوق وی را بر اساس فرمول زیر حساب کنید.

۱۰×تعداد فرزندان + حقوق پایه =حقوق کل

```
Program test;
Var
  Salary:longint;
  Child:byte;
  kole:integer
Begin
  Writeln('enter salary and number of child);
  Readln(salary,child);
  Kole := salary + child *10
  Writeln('kole is ',kole);
END.
```

اولویت عملگرها:

× /

+ -

شرط:

if شرط then

Begin

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

end

else

begin

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

end;

**نکته :** دستورالعمل قبل از IF سمی کالن نمی گیرد

**عناصر شرط:**

< > <= >= = <>

اپراتورها :

AND OR NOT XOR

A	B	AND	OR	XOR
1	1	1	1	1
1	0	0	1	0
0	1	0	1	0
0	0	0	0	1

مثال : برنامه ای بنویسید که با دریافت دو عدد بزرگترین آنها را چاپ کند.

```

Program test;
Var
  a,b:integer;
if a>b then
  begin
    writeln(a);
  end
else
  writeln(b);
end.

```

مثال : برنامه ای بنویسید که با دریافت سه عدد به عنوان ضرایب  $y=ax^2+bx+c$  معادله درجه دو را حل کند(این برنامه را با یک If بنویسید).

```

Var
  A,b,c:integer;
  D,x1,x2:real;
Begin
  Writeln('enter a,b,c');
  Readln(a,b,c);
  D:=b*b-
4*a*c;
  If (d >=0 ) then
    Begin
      X1:=(-b+sqrt(d))/2*a);
      X1:=(-b-sqrt(d))/2*a);

```

مثال : برنامه ای بنویسید که تعداد فرزندان و حقوق پایه و رتبه یک کارگر را از ورودی دریافت و حقوق کل وی را بر اساس فرمول زیر حساب کند.

کسریها - مزایا + بیمه = حقوق کل

مزايا = ۵ × رتبه + ۱۰۰۰ × تعداد فرزند = مزایا

مالیات + بیمه = کسریها

تعداد فرزند	کمتر یا مساوی ۲	بیشتر
بیمه	۱۰۰۰	تعداد فرزندان
رتبه	۰ تا ۲۰	هر چیزی دیگر
مالیات	حقوق پایه × ۱۰/۱۰۰	حقوق پایه

```
begin
writeln('enter salary and grade and number of child');
readln(salary ,num,grade);
if num<=2 then
bimeh:=100
else
bimeh :=num*500;
if (grade>=0) and (grade<=20)then
net := 10/100*salary
else
if grade >20 then
begin
net:=20/100*salary;
mazaya:=num*1000+grade*50;
kasry:=bimeh+net;
kol :=salary + mazaya-kasry;
end;
writeln(kol);
end.
```

تکلیف : در یک ترکیب شیمیایی ۴ عنصر شرکت دارند مقدار مجاز برای تهیه ماده ای به نام asxd به این صورت است.

**A** 0 ~ 50  
**S** 0.5 ~ 0.83  
**X** 81 ~ 92 or 824 ~ 901  
**D** -100 ~ 100

با دریافت مقادیر a,s,x,d از ورودی به ما بگوید که آیا می توان این ماده را تولید کرد یا نه ؟

```
Begin
Num:=0;
Writeln('please enter a s x d');
Readln(a,s,x,d);
If (a>0) and(a<50) then
Num:=num+1;
If (s>0.5)and (s<0.83) then
```

```

Num:=num+1;
If ((x>81)and(x<92))or ((x>824)and(x<901))then
  Num:=num+1;
If (d>-100) and (d<100) then
  Num:=num+1;
If num=4 then
  Writeln('yes can')
Else
  Writeln('you can not');
End.

```

**تکلیف :** برنامه ای بنویسید که با دریافت سه عدد و با فرض اینکه عدد دوم وتر میباشد به ما بگوید که آیا این سه عدد تشکیل یک مثلث قائم الزاویه می دهد یا خیر؟

### :CASE-

CASE میتواند تعدادی شرط را بگیرد و انتخاب کند.

مقادیری که دستور CASE میپذیرد ORDINAL است مقادیری که ORDINAL هستند دارای ترکیب نیستند و همچنین قابل شمارش هم هستند مثل INTEGER, CHARACTER, BYTE, BOOLEAN, LONGINT, ... پس ORDINAL <---- REAL, STRING نیستند

CASE تنها دستوری است که BEGIN ندارد ولی END دارد

مقدارها میتوانند شامل یک مقدار یا چند مقدار که از یکدیگر با کاما جدا شده اند باشند و یا یک سری شمارشی باشند مثل:

```

1:-----
1,10,28:-----
1..100,150..243:-----
'A'..'Z':-----

```

```

Case  متغیر  Of
      مقدار ۱ : Begin
                -----
                -----
                End;
      مقدار ۲ : Begin
                -----
                -----
                End;
      Else:    Begin
                -----
                -----
                End;
End.

```

مثال : برنامه ای بنویسید که یک عدد را از ورودی دریافت ( $0 \leq x < 3$ ) و تلفظ آنها را چاپ کند.

```
BEGIN
READLN(X);
CASE X OF
0:WRITELN('ZERO');
1:WRITELN('ONE');
2:WRITELN('TWO')
ELSE
WRITELN('ERROR');
END;
END.
```

مثال : برنامه ای بنویسید که دو عدد را از ورودی دریافت و چنانچه عدد اول در بازه ۰ تا ۵ بود توان دوم عدد دوم را چاپ کند چنانچه ۶۸ بود حاصلضرب دو عدد را چاپ کند و در غیر اینصورت مقادیر مجاز را چاپ کند.

```
BEGIN
WRITELN('ENTER TWO NUMBER');
READLN(A,B);
CASE A OF
0..5 :WRITELN(B*B);
68:WRITELN(A*B)
ELSE
WRITELN('ENTER 0..5 OR 68');
END;
END.
```

- معادل برنامه بالا با دستور If :

```
Begin
Writeln('enter 2 number');
Readln(a,b);
If (a>=0)and(a<=5) then
Writeln(b*b)
Else
If a=68 then
Writeln(a*b)
Else
Writeln('enter 0..5 or68');
End.
```

تکلیف : روشی برای تلفظ اعداد پیدا کنید.

حلقه تکرار :

۱- تعداد معین



for مقدار نهایی { to مقدار شروع := متغیر  
downto

begin

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

end;

توجه مهم : در برخی از حلقه ها دستورات داخل حلقه به متغیر حلقه مرتبط می باشند و در برخی موارد هیچگونه ارتباطی وجود ندارد.

حلقه for برای تعداد تکرار معین به کار میرود

Downto : در مواردی به کار میرود که مقدار اولیه از مقدار نهایی بیشتر باشد

مثال : برنامه ای بنویسید که توان دوم اعداد ۱۰۰ تا ۲۰۰ را چاپ کند.

```
Begin
For i:=100 to 200 do
Begin
Writeln(i*i);
End;
End.
```

نکته : اجازه ندارید مقدار متغیر حلقه را در داخل بدنه عوض کنید .

مثال : برنامه ای بنویسید که حاصلجمع اعداد ۱..۱۰۰ را چاپ کند.

```
Begin
Sum:=0;
For i:=1 to 10 do
Begin
Sum:=sum+i
End;
Write(sum);
End.
```

مثال : برنامه ای بنویسید که حاصلجمع مضارب ۵ بین ۰ تا ۱۰۰ را به دست آورد.

```
Begin
Sum:=0;
For I:=1 to 20 do
Begin
Sum:=sum+i*5;
End;
Writeln(sum);
```

End.

مثال : برنامه ای بنویسید که دو عدد را از ورودی دریافت و اعداد ما بین این دو عدد را چاپ کند (a<b)

```
Begin
Writeln('enter a,b');
Readln(a,b);
For I:= a to b do
  Writeln(i);
End.
```

مثال : برنامه ای بنویسید که با یک حلقه for اعداد بین ۱۰۰..۱ و ۷۳۸..۳۵۳ و ۹۵۰..۹۴۰ را چاپ کند

```
Begin
For i:=1 to 950 do
Begin
  If ( i>=1) and (i<=100) then
    Writeln(i);
  If (i>353) and (i<=738) then
    Writeln(i);
  If (i>=940) and (i<=950) then
    Writeln(i);
End;
End.
```

معادل برنامه بالا با استفاده از دستور case :

```
Begin
For i:=1 to 950 do
Case I of
  1..100,353..738,940..950: writeln(i);
end;
end.
```

mod → (باقیمانده)

مثال :

X mod y

مثال : برنامه ای بنویسید که مضارب ۷ بین ۱ تا ۱۰۰ را چاپ کند

```
Begin
For i:= 1 to 100 do
Begin
  If I mod 7 = 0 then
    Writeln(i);
End;
End.
```

مثال : برنامه ای بنویسید که تعداد مضارب ۷ و ۵ را در بازه ۲۳۱~۸۴۶ چاپ کند.

Begin

```

K:=0;
C:=0;
For i:=846 downto 231 do
Begin
  If I mod 5 = 0 then
    C:=c+1;
  If I mod 7 = 0 then
    K:=k+1;
End;
Writeln('5=',c,'7=',k);
End;
End.

```

مثال : برنامه ای بنویسید که حقوق پایه و تعداد فرزندان ۱۰۰ کارگر را از ورودی دریافت و مبلغ حقوق آنها را بر اساس فرمول زیر حساب کند .

$$۱۰۰۰ \times \text{تعداد فرزند} + \text{حقوق پایه} = \text{حقوق کل}$$

```

Begin
For i:=1 to 100 do
Begin
Writeln('enter salary and number of children');
readln(salary,num);
kol := salary + num * 1000;
writeln(kol);
end;
end.

```

مثال : برنامه ای بنویسید که ۱۰۰ عدد را از ورودی دریافت و میانگین و حاصلجمع آنها را به ما بدهد.

```

Begin
Sum:=0;
For i:= 1 to 100 do
Begin
Writeln('enter num');
Readln(num);
Sum:=sum+num;
End;
Ave:=ave/100;
Writeln(sum,ave);
End.

```

تعداد تکرار نامعین :

```

while      شرط      Do
  begin
    _____
    _____
    _____
  end;

```

مثال : برنامه ای بنویسید که اعداد ۱..۱۰۰ را چاپ کند

```

Begin
I:=1;
While i<=100 do
Begin
  Write(i);
  I:=i+1;
End;

```

مثال : برنامه ای بنویسید که تعدادی اسم را از ورودی دریافت و به ما بگوید چند بار اسم ali تکرار شده است(آخرین اسم end است).

```

Var
S:integer;
St:string;
Begin
Read(st);
While st<>'end' do
Begin
If st='ali' then
S:=s+1;
Read(st);
End;write(s);
End.

```

مثال : برنامه ای بنویسید که تعدادی عدد که آخرین آنها صفر است را از ورودی دریافت و حاصل جمع آنها را چاپ کند.

```

Begin
Readln(num);
T:=1;
While num<>0 do
  Begin
    Sum:=sum+num;
    Readln(num);
    T:=t+1;
  End;
Writeln('s=',s,'average',s/t);
End;

```

**تکلیف :** برای دانش آموزان یک کلاس برنامه ای بنویسید که نام آنها به همراه تعداد نمرات و سپس نمرات آنها را به فرم زیر از ورودی دریافت کند. تعداد دانش آموزان کلاس نامعین است ولی آخرین نفر نامش end است. برنامه میبایستی پس از دریافت هر یک از اطلاعات پیامی به شکل زیر چاپ کند.

Name: Ali      Sum:      Average:

3  
18  
20  
13.5

```
program test;
var
name:string[10];
ave,sum,nomreh:real;
num,i:integer;
begin
writeln('enter name ');
readln(name);
while name <> 'end' do
begin
writeln('enter num:');
readln(num);
for i:=1 to num do
begin
writeln('enter nomreh: ');
readln(nomreh);
sum:=sum+nomreh;
end;
ave:=sum/num;
writeln('name:',name,'sum:',sum,'ave:',ave);
writeln('enter name ');
readln(name);
end;
end.
```

**تکلیف :** میخواهیم برای برنامه های خود یک منو تهیه کنیم این منو شامل موارد زیر است

1) load  
2) save  
3) edit  
4) exit  
please select a number :

```

program test;
var
num:integer;
begin
writeln(' 1) load');
writeln(' 2) save');
writeln(' 3) edit');
writeln(' 4) exit');
writeln('please select a number : ');
readln(num);
while num<>4 do
begin
case num of
1:writeln('loading');
2:writeln('saving');
3:writeln('editing');
end;
end;
end.

```

مثال : برنامه ای بنویسید که دو عدد را از ورودی دریافت و طبق جدول زیر نسبت به مقدار آن واکنش نشان دهد

a	۱	۲	۳
واکنش	b..1	b..۱	۱+۲+۳+...+B

```

Begin
Writeln (' enter a,b :');
Readln(a,b);
Case a of
1:begin
for i:=1 to b do
write(i);
end;
2:begin
or I:=b downto 1 do
write(i);
end;
3:begin
sum:=0;
for i:=1 to b do
sum:=sum+I;
write(sum);
end;
end.

```

آرایه ها :

آرایه ها به عناصری اطلاق میشود که با یک نام تعدادی محل از حافظه را کنترل کند برای تشخیص محلها از اندیس استفاده میشود

انواع آرایه ها :

۱- آرایه های یک بعدی

۲- عناصر دو بعدی

۳- آرایه های سه بعدی

تعریف آرایه ها :

var

نوع of [اندیس پایان..اندیس شروع] array : نام آرایه

نوع متغیر آرایه : متغیر آرایه

خواندن آرایه :

مثال : برنامه ای بنویسید که نمرات ۱۰۰ دانش آموز را از ورودی دریافت و در یک آرایه بریزد.

Var

A:array[1..100] of real;

Begin

For i:=1 to 100 do

Readln(a[i]);

مثال : برنامه ای بنویسید که نمرات ۱۰۰ دانش آموز را از ورودی دریافت و ۵۰ تای اول را چاپ کند سپس تعداد نمرات ۲۰ را مشخص کند.

Var

A:array[1..100] of real;

Begin

For I:=1 to 100 do

Writeln('enter no(',I,')');

Readln(a[i]);

For i:=1 to 50 do

Writeln(a[i]);

S:=0;

For i:=1 to 100 do

If i:=1 to 100 do

If a[i]=20 then

S:=s+1;

Writeln(s);

End.

پیدا کردن max,min

min = عنصر اول

حلقه برای گردش در آرایه

if  $\text{Min} < \text{عنصر فعلی}$

min := عنصر فعلی

Write(min);

مثال: برنامه ای بنویسید که با دریافت نمرات ۱۰۰ دانش آموز بالاترین و پایین ترین نمره را در بین این ۱۰۰ نفر پیدا کند.

Var

A:array[1..100] of real;

I:integer;

Begin

For i:=1 to 100 do

  Readln(a[i]);

  Min:=a[1];

  Max:=a[1];

  For i:=2 to 100 do

    Begin

      If a[i]< min then

        Min:=a[i];

      If a[i]>max then

        Max := a[i];

    End;

  Writeln('max is ',max,'min is ',min);

End.

مثال: برنامه ای بنویسید که با دریافت ۱۰۰ اسم به ما بگوید تعداد تکرار اسم ali چند بار است و آیا اسم hassan بیشتر ذکر شده یا اسم reza؟

Var

A:array [1..100] of string;

Begin

For i:=1 to 100 do

  Readln(a[i]);

  Ali:=0;

  Reza:=0;

  Hassan:=0;

  For i:=1 to 100 do

    Begin

      If a[i]='ali' then

        Ali:=ali+1;

      If a[i]='reza' then

        Reza:=reza+1;



```

If a[i]='hassan' then
Hassan:=hassan+1;
Endd
Writeln(ali is ':',ali);
If reza>hassan then
Writeln('reza')
Else
Writeln('hassan');
End.

```

مثال : برنامه ای بنویسید که با دریافت نام و نمره ۱۰۰ دانش آموز تنبل ترین فرد کلاس را مشخص کند.

```

Var
Name:array[1..100] of string;
No:array[1..100] of real;
Begin
For i:=1 to 100 do
Begin
Readln(name[i]);
Readln(no[i]);
End;
Min:=no[1];
Namemin:=name[1];
For i:=2 to 100 do begin
If no[i]<min then
Begin
Min:=no[i];
Namemin:=name[i];
Writeln(namemin);
End.

```

آرایه های دو بعدی:

وقتی از آرایه دو بعدی استفاده میکنیم باید از دو حقه for استفاده کنیم.

مثال : برنامه ای بنویسید که نمرات ۴ کلاس که هر یک ۲۰ دانش آموز دارد را از ورودی دریافت و تنبل ترین فرد را در این ۴ کلاس مشخص کند.

```

Var
A:array[1..20,1..4] of real;
Begin
For i:=1 to 4 do
Begin
For j:=1 to 20 do
Readln(a[j,i]);
End;
Min:=a[1,1];
For i:=1 to 4 do
For j:=1 to 20 do

```

```

If a[j,i]< min then
Min:=a[j,i];
Writeln(min);
End.

```

مثال : در یک آرایه  $20 \times 100$  دوهزار کاراکتر نگهداری میشود برنامه ای بنویسید که تعداد تکرار حرف A را با a مقایسه کند و همچنین به ما بگوید آیا کاراکتر Z در این ماتریس وجود دارد ؟

```

Var
A:array[1..20,1..100] of character;
Bool:Boolean;
B,b1:integer;
Begin
For i:=1 to 100 do
For j:=1 to 20 do
                If a[I,j]='A' then
                B:=b+1;
If a[I,j]='a' then
B1:=b1+1;
                If a[I,j]='z' then
Bool:=true;
End;
If b>b1 then writeln('A>a');
If b<b1 then writeln('A<a');
If b=b1 then writeln('A=a');
If true then writeln('"z"exist');
End.

```

تکلیف : میخواهیم نام و نمره تعدادی دانش آموز یک کلاس را از ورودی دریافت کنیم و مشخص کنیم که تنبل ترین و زرنگ ترین فرد کلاس کیست. چند نفر نمره زیر ۱۰ گرفتند؟ نام کسانی که نمره آنها در بازه ۱۷ تا ۱۵ می باشد ب همراه نمره آنها چاپ شود آخرین نفر نامش end است حداکثر تعداد دانش آموزان ۱۰۰ نفر است.

```

Var
a:array [1..100] of string ;
b:array[1..100] of real;
begin
i:=1;
read(a[i]);
while a[i]<>'end' do begin
readln(b[i]);
i:=i+1;
readln(a[i]);
end;
j:=i-1;
min:=a[1];

```

```

max:=a[1];
for i:= 2 to j do
begin
if b[i] >max then
begin
max:=b[i];
name:a[i];
end;
end
writeln(name);
count:=0;
for i:=1 to j do
if b[i]<=10 then
count:=count+1;
writeln(count);
for i:=1 to j do
if (b[i]<17) and (b[i]>15) then
write(b[i],a[i]);
end.

```

**تکلیف :** برنامه ای بنویسید که حاصل این عبارت را حساب کند.

$$i/i! = 1/1! + 2/2! + 3/3! + 4/4! + 5/5!$$

$$N! = 1 * 2 * 3 * 4 * \dots * n$$

```

Begin
Sum:=0;
For n:=1 to 5 do
Begin
For m:=1 to n do
F:=f*m;
Sum:=sum+n/f;
End;
Writeln(sum);
End.

```

**تکلیف :** برنامه ای بنویسید که نمرات ۸۰ دانش آموز را که در چهار کلاس دسته بنده شده اند را از ورودی دریافت کند سپس به سوالات زیر جواب دهد.

۱- معدل هر یک از کلاسها

۲- نمره تنبل ترین و زرنگترین شخص در هر کلاس به طور جداگانه

```

Var
A:array [1..4,1..20] of real;
Begin
For i:=1 to 4 do
For j:=1 to 20 do
Begin
Writeln('please enter nomreh');

```

```

Readln(a[I,j]);
End;
For i:=1 to 4 do
Begin
Sum:=0 ;
For j:=1 to 20 do
Begin
Sum:=a[I,j]+sum;
End;
Writeln(sum/20,'average of this class');
End;
For i:=1 to 4 do
Begin
Max :=a[1,1];
Min:=a[1,1];
For j:=2 to 20 do
Begin
If a[I,j]>max then
Max :=a[I,j];
End;
Writeln(I,min,max);
End;
End.

```

### - String :

**String** آرایه ای از کاراکترهاست. اغلب لازم است کاراکترهایی در رشته موجود درج شوند و یا کاراکترهایی به جای بخشی از رشته نوشته شوند، زیررشته ای در رشته جستجو گردد و یا دو رشته با هم الحاق شوند. **رشته تهی** : طول رشته ثابت نیست و با داده هایی که در آن ذخیره شده است مشخص می گردد. کاراکترهایی که در رشته قرار می گیرد از حداکثر طولی که برای آن تعیین میشود بیشتر نخواهد بود. اگر **name** رشته کاراکتری باشد دستور زیر **name** را به رشته تهی تبدیل میکند:

```
Name:='';
```

### تبدیل رشته به اعداد :

در پاسکال زیر برنامه هایی وجود دارند که به آسانی میتوانند رشته ها را پردازش کنند. زیربرنامه **val** رشته عددی را گرفته به مقدار عددی تبدیل میکند. رشته عددی رشته ای است که حاوی کاراکترهای '0' تا '9' و نقطه ممیز باشد مثل '12.5' و '13.5' زیربرنامه **val** به صورت زیر به کار میرود.

```
Val (st,number,error)
```

**St** رشته ای است که باید به عدد تبدیل شود و **number** متغیری است که رشته تبدیل شده به عدد در آن قرار میگیرد. چنانچه در رشته **st** کاراکتر نا معتبری (کاراکترهای غیر از '0' تا '9' و '!')، محل وجود آن کاراکتر در پارامتر **error**

قرار میگیرد. اگر error برابر با صفر باشد معنایش اینست که کاراکتر نا معتبری وجود ندارد. اگر number از نوع صحیح باشد St به مقدار صحیح و اگر number از نوع اعشاری باشد St به مقدار اعشاری تبدیل میشود. مثال زیر را در نظر بگیرید:

```
Var st :string;
Num,e:integer;
St:='352';
Val (st,num,e);
```

مقدار ۳۵۲ در متغیر num و مقدار صفر در متغیر e قرار میگیرد اکنون دستورات زیر را در نظر بگیرید :

```
St:='35#2';
Val (st,num,e);
```

با اجرای دستور val به دلیل وجود کاراکتر '#' که کاراکتر نامعتبری در اعداد است عدد ۲ در متغیر e قرار میگیرد و معنایش اینست که کاراکتر شماره ۲ ( شماره گذاری از صفر شروع میشود) نامعتبر است و عمل تبدیل صورت نمیگیرد و مقدار num تعریف نشده است.

**تبدیل عدد به رشته :**

برای تبدیل مقدار عددی به رشته عددی از زیربرنامه str استفاده میشود و به صورت زیر به کار میرود:

```
Str (number : format,numstring)
```

با این دستور مقداری که در number قرار دارد به رشته ای به طول format تبدیل میشود و در رشته numstring قرار میگیرد. دستورات زیر را در نظر بگیرید:

```
Var x:integer;
St:string;
X=352;
Str(x,5,st);
```

با اجرای این دستور مقدار 352 به رشته '352' تبدیل شده در st قرار میگیرد

دستورات زیر را در نظر بگیرید:

```
Var y1,y2:real;
St1,st2 : string;
Y1:=352.768
Y2:=476.395
Str(y1:7:2,st1);
Str(y2:3:1,st2);
```

با اجرای دستور str اول مقدار 352.768 به رشته '352.768' تبدیل میشود و در st1 قرار میگیرد و با اجرای دستور str دوم مقدار 476.395 به رشته '476.395' تبدیل میشود و در st2 قرار میگیرد.

جدا کردن زیررشته ای از رشته:

زیررشته بخشی از رشته است و برای جدا کردن زیررشته از رشته از تابع `copy` استفاده میشود تا تابع `copy` به صورت زیر به کار میرود.

`Copy (source,index,size)`

`S:='I am learning pascal ';`

`S1:=copy(s,15,6);`

دستور `copy` باعث میگردد تا با شروع از محل ۱۵ تعداد ۶ کاراکتر از رشته `s` استخراج شود و در `s1` قرار گیرد. بنابراین محتویات رشته `s1` برابر است با `'pascal'`.

**الحاق رشته ها :**

منظور از الحاق رشته ها اتصال رشته ها به یکدیگر است به عنوان مثال اگر `s1:='ab'` و `s2:='cde'` الحاق دو رشته `(s1,s2)` به صورت `'abcde'` خواهد بود برای الحاق رشته ها از تابع `concat` استفاده میشود:

`concat(اسامی رشته ها)`

دستورات زیر را در نظر بگیرید:

`S1:='pascal';`

`S2:='is a ;`

`S3:='language';`

`S4:=concat(s1,s2,s3);`

با اجرای این دستور `s3` به انتهای `s2` متصل میشود و رشته نتیجه به انتهای `s1` متصل میگردد و در نتیجه رشته `s4` عبارت است از: `'pascal is a language'`. اگر طول رشته حاصل بیش از ۲۵۵ باشد بقیه کاراکترها حذف میشوند.

**جستجوی رشته ای در رشته دیگر:**

برای این کار از تابع `pos` استفاده میشود کاربرد این تابع به صورت زیر است:

`Pos(s1,s2)`

`S1` رشته ای است که باید در `s2` وجود داشته باشد محل اولین وقوع آن برگردانده میشود و اگر نه مقدار صفر برگردانده میشود دستورات زیر را در نظر بگیرید:

`S1:='learning'`

`S2:='I am learning pascal';`

`S3:='english'`

`X:=pos(s1,s2)`

`y:=pos(s3,s2)`

چون رشته `s1` در `s2` وجود دارد دستور `pos` اول مقدار ۶ را در `x` قرار میدهد و معنایش این است که رشته `'learning'` در محل ۶ رشته `s2` وجود ندارد مقدار صفر در `y` قرار میگیرد.

**محاسبه طول رشته :**

برای محاسبه طول رشته از تابع `length` به صورت زیر استفاده میشود:

`length(رشته)`

دستورات زیر را در نظر بگیرید:

`S1:='xymn'`

`X:=length(s1)`

چون طول رشته S1 برابر ۴ است مقدار X برابر ۴ خواهد بود.

### حذف و درج زیررشته:

زیر رشته ای را میتوان از رشته ای حذف کرد و یا زیررشته ای را میتوان در رشته ای درج کرد برای حذف زیر رشته از زیربرنامه ها به صورت زیر به کار میروند:

```
Delete(source,index,size)
```

```
Insert(pattern,destination,index)
```

در زیربرنامه delete زیررشته ای به طول size با شروع از محل index از رشته source حذف میشود و در زیربرنامه

insert زیررشته pattern با شروع از محل index در رشته destination درج میشود دستورات زیر را در نظر بگیرید:

```
S1:=pas***cal
```

```
Delete(s1,4,3)
```

```
S2:='paal';
```

```
S3:='sc';
```

```
Insert(s3,s2,);
```

دستور delete باعث میشود تا با شروع از محل ۴ رشته s1 حذف شده و رشته s1 به pascal تبدیل شود دستور

insert موجب میشود تا رشته s3 در s2 درج شود و در نتیجه رشته s2 به 'pascal' تبدیل شود.

مثال: برنامه ای بنویسید که یک اسم را از ورودی دریافت و آنرا بر عکس چاپ کند

```
Readln(name);
```

```
For i:=length(name) downto 1 do
```

```
Write (name[i])
```

مثال: برنامه ای بنویسید که یک نام را از ورودی دریافت و به ما بگوید که آیا این نام با حرف a شروع میشود یا

خیر؟

```
Readln(name);
```

```
If name[1]='a' then
```

```
Writeln('ok')
```

```
Else
```

```
Writeln('not ok');
```

مثال: برنامه ای بنویسید که یک نام را از ورودی دریافت و حروف آنرا یک در میان چاپ کند.

```
Readln(name);
```

```
For i:=1 to length(name) do
```

```
If I mod 2 = 0 then
```

```
Writeln(name[i]);
```

```
For i:=1 to int(length(name)/2) do
```

روش دیگر:

```
Writeln(name[i*2]);
```

مثال: برنامه ای بنویسید که یک رشته را از ورودی دریافت و متقارن بودن آنرا چک کند.

```
۱ ۲ ۳ ۴ ۵ ۵ ۴ ۳ ۲ ۱
```

```
var
```

```
N:string;
```

```

Begin
Readln(n);
For i:=1 to int(length(n)/2) do
Begin
If n[i] <> n[length(n)-i+1] then
K:=0;
End;
If k=1 then
Writeln ('ok')
Else
Writeln('no');
End.

```

مثال : برنامه ای بنویسید که تعداد حروف a موجود در یک رشته ورودی را بشمارد.

```

Var st:string;
Begin
Readln(st);
C:=0;
For i:=1 to length(st) do
If st[i]='a' then →if st[i] in ['a','A'] then
C:=c+1;
Writeln(c)

```

مثال : برنامه ای بنویسید که تعداد اسامی alireza موجود در رشته را بشمارد.

```

Begin
Readln(st);
C:=0;
While pos('alireza',st) <> 0 do
Begin
C:=c+1;
J:=pos('alireza',st);
Delete(st,pos('ali',st),3);
End;
Write ( c );
End.

```

مثال : برنامه ای بنویسید که دو رشته را از ورودی دریافت و بعد از کاراکتر مساوی که در رشته اول وجود دارد رشته دوم را چاپ کند.

```

Readln(st,st1);
J:=pos('=',st);
Writeln(copy ((st,1 j) , st1, copy (st,j+1,100)));

```

تکلیف : برنامه ای بنویسید که با دریافت سه حرف و قرار دادن آنها در یک آرایه سه تایی کلیه ترکیبات ممکن را که سه حرف ترکیباتشان به هم نخورد چاپ کند.

```

W a x
A x w

```



W a x

مثال : برنامه ای بنویسید که یک رشته را از ورودی دریافت و کلیه کلمات موجود در آن را بشمارد.

```

Readln(st);
S:=0;
St:=st+' '
While pos(' ',st) <> 0 do
Begin
S:=s+1;
Delete(st,1,pos(' ',st));
While st[1]=' 'do
Delet (st,1,1);
End;
End.

```

تکلیف : برنامه ای بنویسید که ۱۰۰ رشته را از ورودی دریافت و در یک آرایه به طول ۱۰۰ از نوع string بریزید و به

سؤالات زیر جواب دهد.

- ۱- تعداد کل کلمات
- ۱- تعداد کل حروف
- ۲- تعداد حروف صدا دار

ذخیره اطلاعات :

Log file : فایلی است که تمام تغییرات مربوط به یک محیط را ثبت میکند

فایلها :

- ۱- متنی text
- ۲- رکوردی typed
- ۳- بدون نوع
- ۴- untyped

معرفی فایل‌های متنی :

Var

Text : نام فایل

نسبت دادن فایل :

assign ('نام خارجی', نام فایل) ;

مثال :

```
Assign(f,'c:\a1.dat.ddd');
```

باز کردن جهت خواندن

```
reset(نام فایل);
```

باز کردن جهت نوشتن

```
rewrite(نام فایل);
```

```
readln(نام فایل, ___)
```

```
writeln(نام فایل, ___)
```

```
close(نام فایل)
```

تا `close` انجام نشود `data` ذخیره نمیشود .

نکته بسیار مهم : در هر یک از مسائلی که در مورد فایلها مطرح میشود می بایستی به نحوی از یکی از تکنیکهای نگهداری اطلاعات در حافظه اصلی استفاده نمود این تکنیکها ممکن است استفاده از متغیرها و آرایه ها و ماتریسها و درختها... استفاده نمود ولی تنها با دو عمل خواندن و نوشتن به روی فایل کار انجام میشود

`Update` : میخوانیم ولی دوباره میریزیم سر جاش

`Append` : بهش یک چیزی اضافه میکنیم

نکته : عمل `rewrite` باعث میگردد چنانچه فایل وجود نداشته باشد ایجاد و چنانچه وجود دارد اطلاعاتش به طور کامل پاک شود.

مثال : برنامه ای بنویسید که ۱۰۰ اسم را از ورودی دریافت و آنها را در یک فایل به نام `(a1.dat)` بنویسد.

```
Var
F:text;a:string;
Begin
Assign(f,'a1.dat');
Rewrite(f);
For i:=1 to 100 do
Readln(a);
Writeln(f,a);
End;
Close(f);
End.
```

مثال : برنامه ای بنویسید که فایل `a1.dat` را خوانده و به ما بگوید چند بار اسم `ali` تکرار شده است؟

```
Var
A:text;
B:string;
Begin
Assign(a,'a1.dat');
Reset(a);
Sum:=0;
For i:=1 to 00 do
Begin
```

```

Readln(a,b);
If b='ali' then
    Sum:=sum+1;
End;
Close(a,b);
End;

```

مثال : برنامه ای بنویسید که تعدادی اسم را که آخرین آنها **end** است از ورودی دریافت و در یک فایل به نام **aa.dat** بریزد آنگاه فایل را بسته و قسمتهای زیر را به طور جداگانه انجام دهد.

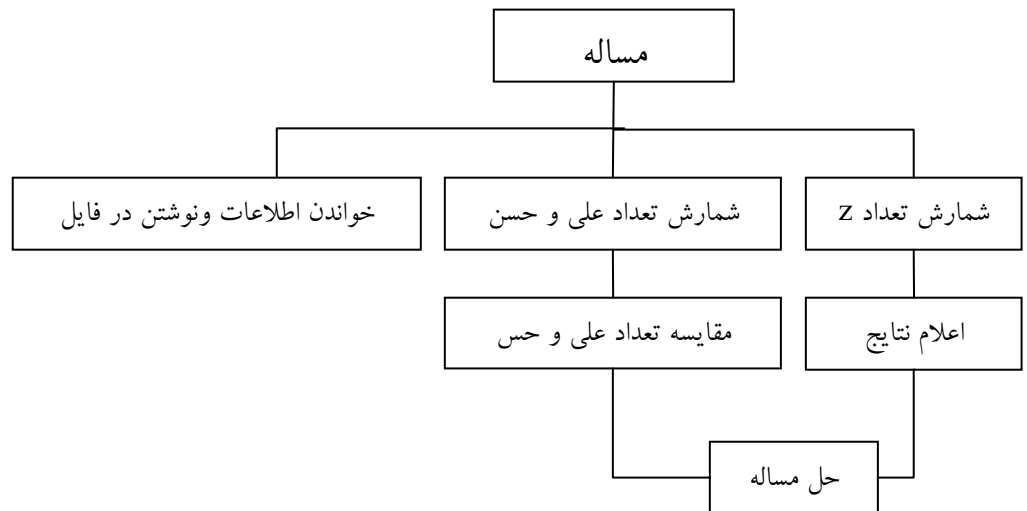
۱- تعداد حسنها بیستر است یا علی ها

۲- چند اسم وجود دارد که با حرف Z شروع میشود.

```

Var
f:text;
Name:string;
Begin
Assign(f,'aa.dat');
Rewrite(f);
Readln(name);
While name<>'end' do
Begin
Writeln(f,name);
Readln(name);
End;
Close(f);
H:=1;
A:=1;
Z:=1;
Reset(f);
While not eof (f) do
Begin
Readln(f,name);
If name='ali' then
A:=a+1;
If name ='hassan' then
H:=h+1;
End;
Close(a);
If h>a then writeln('h>a');
If h<a then writeln('h<a');
If h=a then writeln('h=a');
End;
Close(f);
Reset(f);

```



```

While not eof (f) do
Begin
Readln(f,name);
If name[i]='z' then
Z:=z+1;
End;
Close(f);
Writeln('sum of z is :z');
End.

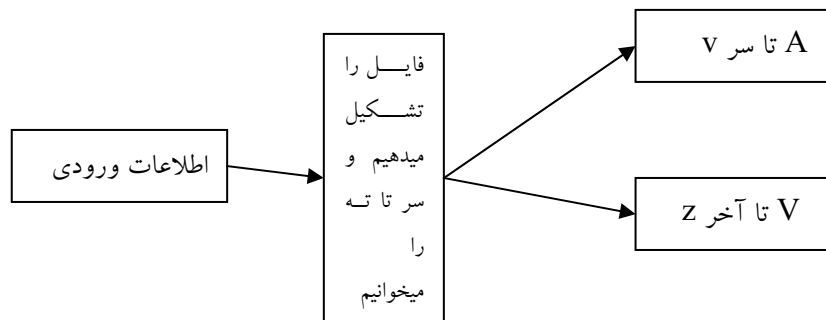
```

**تکلیف:** برنامه ای بنویسید که تعدادی نام را از ورودی دریافت و در یک فایل بریزد سپس فایل تشکیل شده را باز کرده و از روی این فایل دو فایل دیگر تشکیل دهید که در یکی از آنها اسامی که بین a تا z قرار گرفته اند ریخته و در فایل دوم کلیه اسامی که از v تا z هستند را بریزد.

```

Program test;
Var
Name : string;
F,f1,f2:text;
Begin
Assign(f,'a.dat');
Rewrite(f);
Writeln('enter a name ');
Readln(name);
While length(name)>0 do
Begin
Writeln(f,name);
Writeln('enter a name');
Readln(name);
End;
Close(f);
Reset(f);
Assign(f1,'a1.dat');
Assign(f2,'a2.dat');
Rewrite(f1);
Rewrite(f2);
While not eof (f) do
begin
Readln(f,name);
Case name[1] of
'a'..'u':writeln(f1,name);
'v'..'z':writeln(f2,name);
end;
end;
close(f2);
close(f1);
close(f);

```



end.

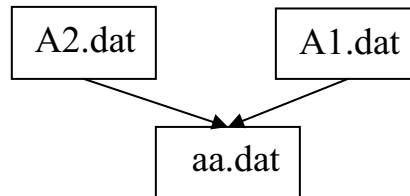
**تکلیف:** برنامه ای بنویسید که نام دو فایل را از ورودی دریافت و از اطلاعات داخل این دو فایل فایل سومی تشکیل دهد که حاصل ترکیب دو فایل قبل باشد.

```

var
h,f,g:text;
a:string;
begin

writeln('enter first filename') ;
readln(a);
assign(h,a);
writeln('enter second filename') ;
readln(a);
assign(f,a);
reset(h);
reset(f);
assign(g,'out.dat');
rewrite(g);
while not eof(h) do
begin
readln(h,a);
writeln(g,a);
end;
while not eof(f) do
begin
readln(f,a);
writeln(g,a);
end;
close(f);
close(h);
close(g);
End.

```



**تکلیف:** برنامه ای بنویسید که در ضمن عمل merge دو فایل مرتب شده را با یکدیگر ترکیب و یک فایل مرتب شده به دست آوریم.