

نمونہ مسائل حل شدہ برنامہ نویسی C

(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.

```
#include<stdio.h>
void main()
{ int i=999;
  while ( i>100 )
  {
    printf("%d,",i);
    i-=2;
  }
}
```

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

۲. * برنامه ای را بنویسید که دو عدد صحیح را بگیرد و تمام 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 دیگری بوسیله کاربر وارد می شود را بخواند یا دریافت کند.

```
#include<stdio.h>
void main()
{ int a, b, c;

  printf("Enter 2 numbers :");
  scanf("%d%d", &a, &b);
  if (a>b)
  { c=a;
    a=b;
    b=c;
  }
  if (a%2 !=0)
    a--;
  while (b>a+2)
  { a+=2;
    printf("%d ",a);
  }
}
```

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 a و عدد بزرگتر را در ظرف y قرار می دهد. سپس با استفاده از یک حلقه *while* تعریف کرده ایم که تا زمانی که $a < b$ است کارهای قسمت دو را به شرح ذیل انجام دهد:

اگر a بر 2 بخش پذیر باشد (شرط برقرار باشد) یعنی a عددی زوج است پس از بعد از آن را چاپ کن (با استفاده از دستور $a+=2$ هر بار 2 تا به مقدار x می افزاید) تا b ، والا یک واحد از a 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.

```
#include<stdio.h>
void main()
{
    int i,n;
    double sum=0, avg, mark;

    printf("enter n of students:");
    scanf("%d", &n);
    for (i=1; i<=n; i++)
    {
        printf("enter a mark :");
        scanf("%lf", &mark);
        sum+=mark;
    }
    avg=sum/n;
    printf("\naverage=%5.2lf", avg);
}
```

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.

۳

```
#include<stdio.h>
void main()
{
    int max=-32767, min=32767, a, i;
    for (i=1; i<=100; i++)
    {
        printf("enter a number :");
        scanf("%d", &a);
        if(a<min)
            min=a;
        if(a>max)
            max=a;
    }
    printf("max is:%d \n min is:%d ", max, min);
}
```

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.

```
#include<stdio.h>
void main()
{
    int i, a, b, bmm, kmm;
    printf("enter 2 numbers:");
    scanf("%d%d", &a, &b);
    for (i=a; i>=1; i--)
        if(a%i==0 && b%i==0)
        {
            bmm=i;
            break;
        }
    kmm=a*b/bmm;
    printf("bmm=%d, kmm=%d", bmm, kmm);
}
```

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 در ۱۰ را چاپ کند.

```
#include<stdio.h>
void main()
{
    int i, j;
    for (i=1; i<=10; i++)
    {
        printf("\n");
    }
}
```

```

for(j=1; j<=10; j++)
    printf("%3d ", i*j);
}
}

```

نکته ای که حائز اهمیت است اینست که ابتدا حلقه داخلی به ازای هر i ، $n=j$ یعنی 10 بار اجرا می شود و بعد مقداری به اضافه می کند و دوباره ...

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

$$1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} - \frac{1}{6} + \dots$$

برای حل این گونه مسائل چه در برنامه نویسی و چه در ریاضیات باید ابتدا معادله کلی عبارت را بیابیم که با کمی دقت می توان جمله عمومی زیر را برای آن در نظر گرفت:

$$\frac{(-1)^n}{n+1}$$

به ازای $n=0$ عبارت برای تک تک جملات برقرار است. حال کافی است این جمله برای کلیه جملات مورد محاسبه قرار گیرد و در ظرفی هر بار حاصل جمله جدید با جملات قبل جمع شود.

که این عمل را با استفاده از یک حلقه `for` انجام می دهیم.

البته هر چند در این جا مطرح نشده که این عبارت را تا چه مقداری محاسبه کنیم لذا با تعریف کردن متغیر n دست کاربر را برای محاسبه عبارت باز نموده ایم.

البته برنامه طوری نوشته شده است که عدد وارد شده باید بزرگتر از 1 باشد. البته اگر عدد 1 را وارد کند باز برنامه جواب صحیح را به ما می دهد. 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.

```

#include<stdio.h>
void main()
{
    Int i, n;
    double alamat=1, sum=0;
    printf("enter a number:");
    scanf("%d", &n);
    for (i=1 ;i<=n-1; i++)
    {
        sum+=alamat/i;
        alamat=-alamat;
    }
    printf("sum=%.2f", sum);
}

```

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

$$\frac{2 \times 4 \times 6 \times 8 \times \dots \times 100}{1 \times 3 \times 5 \times 7 \times \dots \times 99}$$

```
#include<stdio.h>
void main()
{ int i=2;
  double hasel=1;
  while ( i<=100 )
  {
    hasel*=(1.0*i)/(i-1);
    i+=2;
  }
  printf("hasel=%6.2f", hasel);
}
```

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

```
#include<stdio.h>
void main()
{ int i=0;
  double x;
  printf("enter a number:");
  scanf("%lf", &x);
  for(i=0; i<=x; i++)
  ;
  printf("[%g]=%d", x, i-1);
}
```

۱۰ - ** برنامه بنویسید که عدد صحیحی را بگیرد و با کمترین تعداد دفعات تکرار اول بودن یا

نبودن آن را مشخص کند.

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

```
#include<stdio.h>
#include<math.h>
void main()
{
  int a, k, i;
  printf("enter a number:");
  scanf("%d",&a);
  k=0;
  for (i=2; i<=sqrt(a); i++)
    if(a%i==0)
    {
      printf("%d avval nist", a);
      k=1;
      break;
    }
}
```

```

if (k==0)
    printf("%d avval ast", a);
else
    printf("%d avval nist", a);

}

```

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

برای انجام این کار باید کارهای زیر در یک حلقه ای که از اولین عدد دو رقمی شروع و به آخرین رقم دو رقمی ختم می شود 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 می کند 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 کار خود را تا آخرین عددش ادامه می دهد.

```

#include<stdio.h>
void main()
{
    int a, b, i;
    for (i=10; i<=99; i++)
    {
        a=i/10;
        b=i%10;
        if(i%(a+b)==0)
            printf("%02d", i);
    }
}

```

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

$$\text{hasel} = X^n$$

```

#include<stdio.h>
void main()
{
    int i, n;
    double x, hasel=1;
    printf("enter a double number:");
    scanf("%lf", &x);
    printf("enter an integer number:");
    scanf("%d", &n);
}

```

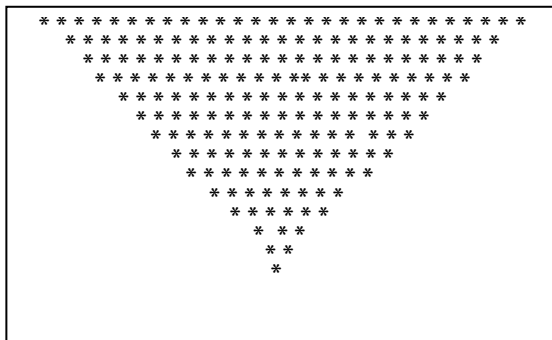

۸

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.

```
#include<stdio.h>
void main()
{ int i, j;

for(i=1; i<=13; i++)
{ for(j=1; j<=i; j++)
printf("*");
printf("\n");
}
for(i=1; i<=12; i++)
{ for(j=12; j>=i; j--)
printf("*");
printf("\n");
}
}
```

(ب)



```
#include<stdio.h>
void main()
{ int a, b, c=25 ;

for(a=1; a<=25; a++)
{ for(b=1; b<=25; b++)
{ if((b<c-a) && (b>=a))
printf("*");
else
printf(" ");
}
printf("\n");
}
}
```

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

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
.	.		.	.
.	.		.	.
.	.		.	.
.	.		.	.

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

$$x + e^x = 5$$

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.

```
#include<stdio.h>
#include<math.h>
void main()
{ double xp, xn, xm;
  xn=0;
  xp=3;
  do
  {
    xm=(xp+xn)/2;
    if(xm+ exp(xm) -5>0)
      xp=xm;
    else
      xn=xm;
  }
  while((xp-xn)>1E-5);
  printf("answer is : %.5f", xm);
}
```

مثالهای آرایه ها

۱۶- ** برنامه ای بنویسید که یک لیست (آرایه یک بعدی) از اعداد و سپس یک عدد را بگیرد و تعداد دفعات تکرار آن عدد را در لیست معین کند .

```
#include<stdio.h>
void main()
{ int i, n=0, num[100], x;
  printf("enter 100 numbers:");
  for(i=0; i<100; i++)
    scanf("%d",&num[i]);
  printf("enter a number:");
  scanf("%d",&x);
  for(i=0; i<100; i++)
    if(x==num[i])
      n++;
  printf("\ntedad=%d",n);
}
```

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- ** برنامه ای بنویسید که یک لیست از اعداد را بگیرد و به ترتیب صعودی مرتب کند.

```
#include<stdio.h>
void main()
{ int i, j, num[10], tmpnum ;
  printf("enter 10 numbers:");
  for(i=0; i<10; i++)
    scanf("%d",&num[i]);
  for(j=0; j<9; j++)
    for(i=0; i<10-j; i++)
      if(num[i]>num[i+1])
        { tmpnum=num[i];
          num[i]=num[i+1];
          num[i+1]=tmpnum;
        }
}
```

```

printf("\n adad moratab shode be tartibe souidi\n");
for(i=0; i<10; i++)
    printf("%d ",num[i]);
}

```

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 با هم مقایسه شدند، بزرگترین عنصر به انتهای آرایه می رود. دفعه بعد که عناصر آرایه دو به دو با هم 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.

```

#include<stdio.h>
void main()
{
    int q[1000], i, j, z=0, n;
    for(i=1; i<1000; i++)
    {
        n=0;
        for(j=1; j<=i; j++)
            if(i%j==0)
                n++;
        if(n==2)
        {
            q[z]=i;
            z++;
        }
    }
    for(i=0; i<z; i++)
        printf("%3d ",q[i]);
}

```

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

```

#include<stdio.h>
void main()
{
    int i;
    double x[100], max, min, avg, v=0, sum=0;
    max=-32767;
    min=32767;
    printf("enter 100 numbers:");
    for(i=0; i<100; i++)

```

```

{ scanf("%lf",&x[i]);
  if(max<x[i])
    max=x[i];
  if(min>x[i])
    min=x[i];
  sum+=x[i];
}
avg=sum/100;
for(i=0;i<100;i++)
  v+=((x[i]-avg)*(x[i]-avg))/100;
printf("\n max=%.2f, min=%.2f", max, min);
printf("\n average=%.2f, variance=%.2f", avg ,v);
}

```

۲۰ - ** برنامه ای بنویسید که لیستی صدتایی از اعداد را بگیرد و سپس اعداد مثبت آن را و اعداد منفی آن را در آرایه های دیگری ذخیره و جداگانه چاپ کند.

```

#include<stdio.h>
void main()
{
  int i, npos=0, nneg=0;
  double x, neg[100], pos[100];
  printf("\n Enter 100 numbers:");
  for(i=0; i<100; i++)
  {
    scanf("%lf", &x);
    if(x>=0)
    {
      pos[npos]=x;
      npos++;
    }
    else
    {
      neg[nneg]=x;
      nneg++;
    }
  }
  printf("\nThere are the positive numbers:");
  for(i=0; i<npos; i++)
    printf("\n%g", pos[i]);
  printf("\nThere are the negative numbers:");
  for(i=0; i<nneg; i++)
    printf("\n%g", neg[i]);
}

```

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

برای محاسبه دترمینان یک ماتریس 3×3 یک آرایه ۹ تایی برای ذخیره مقدار نه عضو ماتریس کافیت. 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.

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

```
#include<stdio.h>
void main()
{
    int i, j, x1, x2, x3, det, m[3][3];
    for(i=0; i<3; i++)
        for(j=0; j<3; j++)
            scanf("%d", &m[i][j]);
    det = (m[0][0] * (m[1][1] * m[2][2] - m[1][2] * m[2][1])) -
          (m[0][1] * (m[1][0] * m[2][2] - m[1][2] * m[2][0])) +
          (m[0][2] * (m[1][0] * m[2][1] - m[2][0] * m[1][1]));
    printf("determinan=%5.2d", det);
}
```

۲۲ - ** برنامه ای بنویسید که ابتدا n و سپس عناصر یک ماتریس $n \times n$ را بگیرد، مجموع تمام عناصر و نیز حاصلضرب عناصر قطر اصلی را حساب و چاپ کند.

```
#include<stdio.h>
void main()
{
    int i, j, n;
    double sum=0, haqa=1, m[20][20];
    scanf("%d", &n);
    for(i=0; i<n; i++)
        for(j=0; j<n; j++)
            {
                scanf("%lf", &m[i][j]);
                sum+=m[i][j];
            }
    for(i=0; i<n; i++)
        haqa*= m[i][i];
    printf("majmooe hame=%g, haselezarbe anasere qotre asli=%g",
        sum, haqa);
}
```

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

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[0]` (مقدار اولین عدد از لیست اول $i=0$) 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 $i=99$ ادامه می یابد.

```
#include<stdio.h>
void main()
{
    int i, j, n=0, x[100], y[100];
    printf("enter numbers of first list:");
    for(i=0;i<100;i++)
        scanf("%d", &x[i]);
    printf("enter numbers of second list:");
    for(i=0;i<100;i++)
        scanf("%d", &y[i]);

    for(i=0; i<100; i++)
        for(j=0; j<100; j++)
            if(x[i] == y[j])
                n++;
    printf("\n eshtarak=%d", n);
}
```

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

```
#include<stdio.h>
void main()
{
    int i, j, kh[10][10];
    for(i=1; i<=10; i++)
    {
        kh[i][1]=1;
        kh[i][i]=1;
        for(j=1; j<=i-2; j++)
            kh[i][j+1]=kh[i-1][j]+kh[i-1][j+1];
    }
    for(i=1; i<=10; i++)
    {
        printf("\n");
        for(j=1; j<=i; j++)
            printf("%5d", kh[i][j]);
    }
}
```

۲۵- ** برنامه ای بنویسید که رشته ای را بگیرد و معکوس آن را چاپ کند.

```
#include<stdio.h>
void main()
{
    char s[50], i;
    scanf("%s", s);
    for(i=0; s[i]!=NULL; i++)      /* جستجو به دنبال اندیس آخر رشته */
        ;
    for( ; i>0; --i)      /* به عملگر یک واحد کاهش پیشوندی توجه کنید */
        printf("%c", s[i]);
}
```

۲۶- ** برنامه ای بنویسید که دو رشته را بگیرد و آنها را از حیث الفبا مقایسه و نتیجه را چاپ کند (بدون استفاده از تابع کتابخانه ای).

```
#include <stdio.h>
void main()
{
    char s[60], t[60];
    int i, flag=0;
    printf("\n enter s, t :");
    scanf("%s %s", s, t);
    for(i=0; s[i]!=NULL && t[i]!=NULL; i++)
        if(s[i]!=t[i])
            break;
    if(s[i]>t[i])
        printf(" s moqaddam ast ");
    else
        if(s[i]<t[i])
            printf(" t moqaddam ast ");
        else
            printf(" s , t yeksan hastand ");
}
```

۲۷- ** برنامه ای بنویسید که رشته ای را از کاربر بگیرد و آن رشته را به استثنای حروف صدا دارش در رشته دیگری ذخیره نموده و چاپ کند.

```
#include <stdio.h>
void main()
{
    char s1[60], s2[60];
    int i, j=0;
    printf("\nEnter a string\n");
```

```

scanf("%s", s1);
gets(s1);
for(i=0; s1[i]!=NULL; i++)
    if(s1[i]!='a' && s1[i]!='e' && s1[i]!='u' && s1[i]!='y'
        && s1[i]!='o' && s1[i]!='i')
        {
            s2[j]=s1[i];
            j++;
        }
s2[j]=NULL;
printf("\n %s", s2);
}

```

۲۸- **** برنامه ای بنویسید که عددی را در مبنای ۱۶ از کاربر دریافت کرده و مبنای ۱۰ آن را حساب کرده و در خروجی چاپ کند.

عدد مبنای ۱۶ ورودی را باید بناچار به شکل رشته ای دریافت کرد چون ممکن است ارقام حرفی نیز داشته باشد. این ارقام حرفی را بعد از دریافت به شکل الفبای بزرگ تبدیل می کنیم.

```

#include <stdio.h>
void main()
{
    char snum[30];
    int i, j, digit;
    long int hasel=0;
    printf("\nEnter a Hexadecimal number:");
    scanf("%s", snum);
    for(i=0; snum[i]!=NULL; i++)
    {
        if(snum[i]>='a' && snum[i]<='z') /* if char is a small letter */
            snum[i] -= ('a'-'A'); /* convert char to uppercase */
        if(snum[i]>='0' && snum[i]<='9') /* convert ascii code to value */
            digit=snum[i]-'0';
        else
            digit=snum[i]-'A'+10;
        hasel *= 16; /* converting from hex to decimal */
        hasel += digit;
    }
    printf("\n%d", hasel);
}

```

۲۹- **** برنامه ای بنویسید که دو ماتریس را از کاربر گرفته و حاصل ضرب آنها را در ماتریس سوم ذخیره کرده و چاپ کند.

```

#include <stdio.h>
void main()
{

```

```

double a[10][10], b[10][10], c[10][10], t; /* a × b ---> c */
int i, j, k, m, n, p;
/* a is m x n , b is n x p , c will be m x p */
printf("\n Enter dimensions: m, n, p:");
scanf("%d %d %d", &m, &n, &p);
printf("\n Enter elements of matrix a: ");
for(i=0; i<m; i++)
    for(j=0; j<n; j++)
        {
            scanf("%lf", &t);
            a[i][j]=t;
        }
printf("\n Now enter elements of matrix b: ");
for(i=0; i<n; i++)
    for(j=0; j<p; j++)
        {
            scanf("%lf", &t);
            b[i][j]=t;
        }
/* calculating a × b at c ... */
for(i=0; i<m; i++)
    for(j=0; j<p; j++)
        {
            c[i][j]=0;
            for(k=0; k<n; k++)
                c[i][j] += a[i][k] * b[k][j];
        }

/* printing c... */
printf("\n a x b = \n");
for(i=0; i<m; i++)
    {
        printf("\n");
        for(j=0; j<p; j++)
            printf(" %8.3f", c[i][j]);
    }
}

```

۳۰- *** برنامه ای بنویسید که تعدادی اسم (حداکثر ۵۰ حرفی) را تا رسیدن کلمه "finish" دریافت کند و آنها را به ترتیب الفبایی زیر هم چاپ کند و کوتاهترین اسم را نیز مشخص کند.

```

#include <stdio.h>
#include <string.h>
void main()
{
    char names[200][50], shortest[40], stmp[40];
    int n, i, j, minlength=40;

```

```
for(n=0; ; n++)
{
    printf("\nEnter next name:");
    scanf("%s", &names[n]);
    if(strcmp(names[n], "finish"))
        break;
    if(strlen(names[n])<minlength)
    {
        strcpy(shortest, names[n]);
        minlength=strlen(names[n]);
    }
}
printf("\nThe shortest name is %s ", shortest);
for(j=0; j<n; j++) /* Bubble sort */
    for(i=1; i<n; i++)
        if(strcmp(names[i], names[i-1])>0)
        {
            strcpy(stmp, names[i]);
            strcpy(names[i], names[i-1]);
            strcpy(names[i-1], stmp);
        }
printf("\n Sorted names are below: ");
for(i=0; i<n; i++)
    printf("\n%s", names[i]);
}
```

مثالهای متفرقه

۳۱- ** برنامه ای که مقدار x را بر حسب رادیان (x مثبت است) از ورودی خوانده، سینوس آن را از بسط محاسبه کند (تا ۵ رقم اعشار):

$$\sin(x) = x - x^3/3! + x^5/5! - x^7/7! + \dots$$

از رابطه هر جمله با جمله قبل استفاده می کنیم:
جمله قبلی * $-x*x/(n*(n-1)) =$ جمله فعلی

```
#include<stdio.h>
void main()
{ int n=3;
  double x, t, s, abst;
  printf("enter a number:");
  scanf("%lf", &x);
  s=x;
  t=x;
  do
  { t*=-x*x/(n*(n-1));
    s+=t;
    n+=2;
    if (t<0)
      abst=-t;
    else
      abst=t;
  }
  while( abst > 1e-5 );
  printf("sin(%g=%0.5f", x, s);
}
```

۳۲- * برنامه ای که عناصر ماتریس 3×5 را از ورودی خوانده، ترانهاده آن را پیدا می کند و آن را به خروجی می برد. ترانهاده ماتریس، ماتریسی است که سطرهای آن، ستونهای ماتریس اولیه اند.

```
#include<stdio.h>
void main()
{ int i, j, a[3][5], b[5][3];
  for(i=0; i<3; i++)
    for(j=0; j<5; j++)
      { scanf("%d",&a[i][j]);
        b[j][i]=a[i][j];
      }
}
```

```

۲.
for(i=0; i<5; i++)
{ printf("\n");
  for(j=0; j<3; j++)
    printf("%6d", b[i][j]);
}
}

```

۳۳- *** برنامه ای بنویسید که دو عدد n و r را بخواند و مقدار C_n^r محاسبه و چاپ کند:

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

```

#include<stdio.h>
void main()
{ int i, n, r, surat=1, makh=1;
  double hasel;
  scanf("%d%d",&n, &r);
  for(i=n;i>=r+1;i--)
    surat*=i;
  for(i=n-r;i>=1;i--)
    makh*=i;
  hasel=surat/makh;
  printf("hasel=%.0lf",hasel);
}

```

۳۴- *** برنامه ای 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. (مبنا کمتر از ده باشد)

```

#include<stdio.h>
void main()
{ int i=0, j, n, n2, mabna, r, x[50];
  printf("enter a number and base:");
  scanf("%d%d", &n, &mabna);
  n2=n;
  do
  { r=n% mabna;
    n/=mabna;
    x[i]=r;
    i++;
  }
  while(n>= mabna);
}

```

۲۱

```
x[i]=n;
printf("%d in base %d = ", n2, mabna);
for(j=i; j>=0; j--)
    printf("%1d", x[j]);
}
```

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
