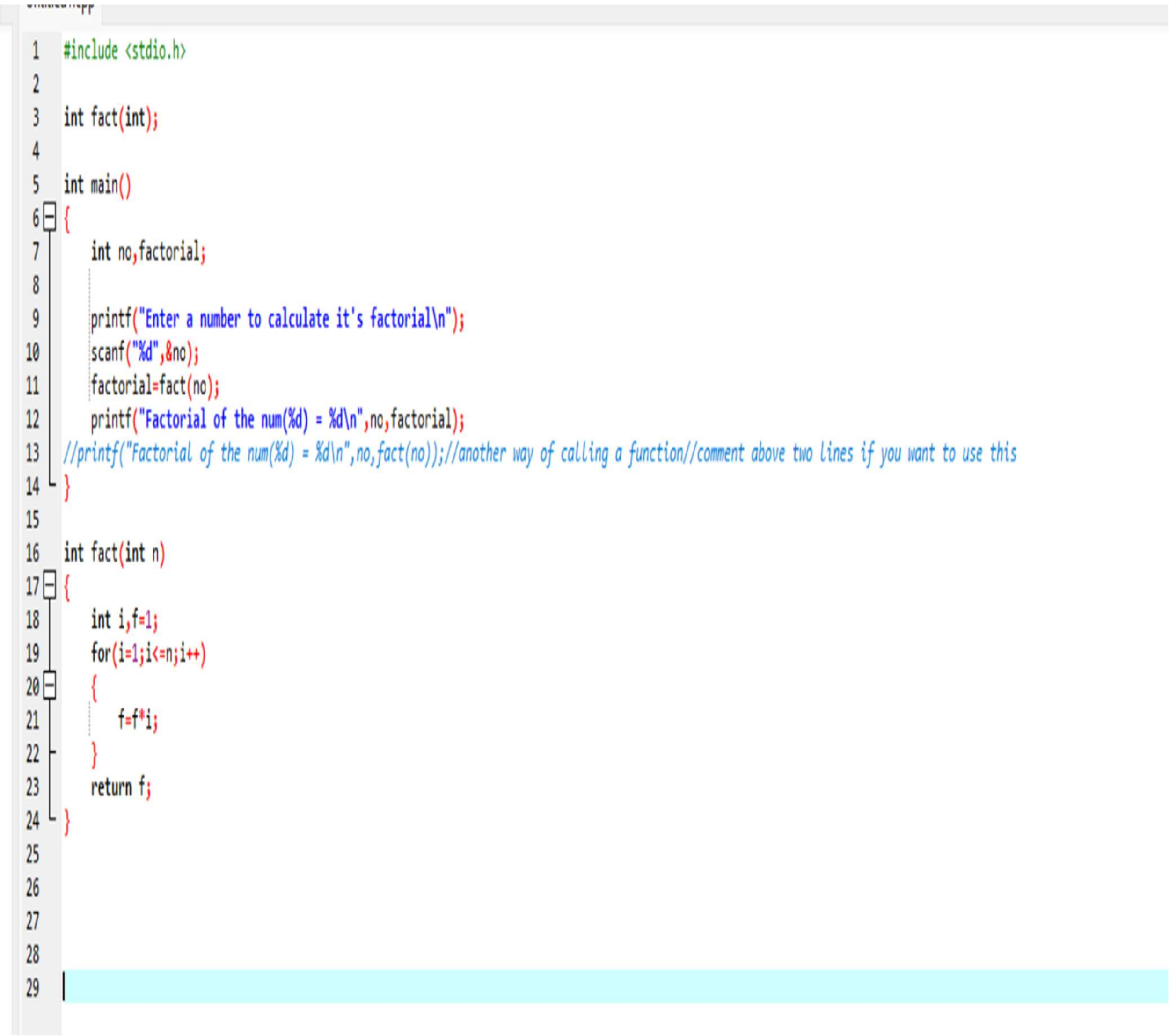
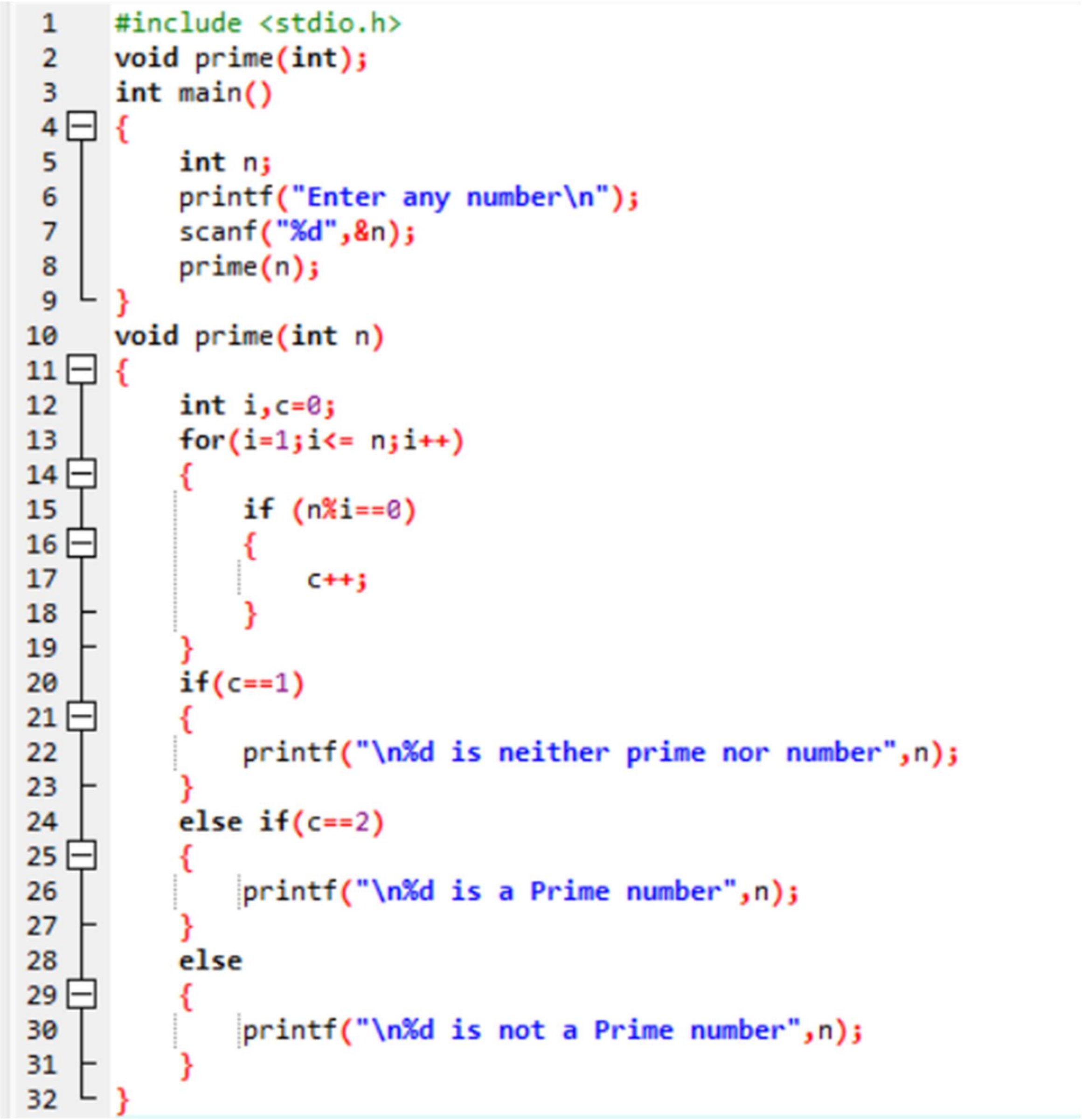
FACTORIAL USING FUNCTION



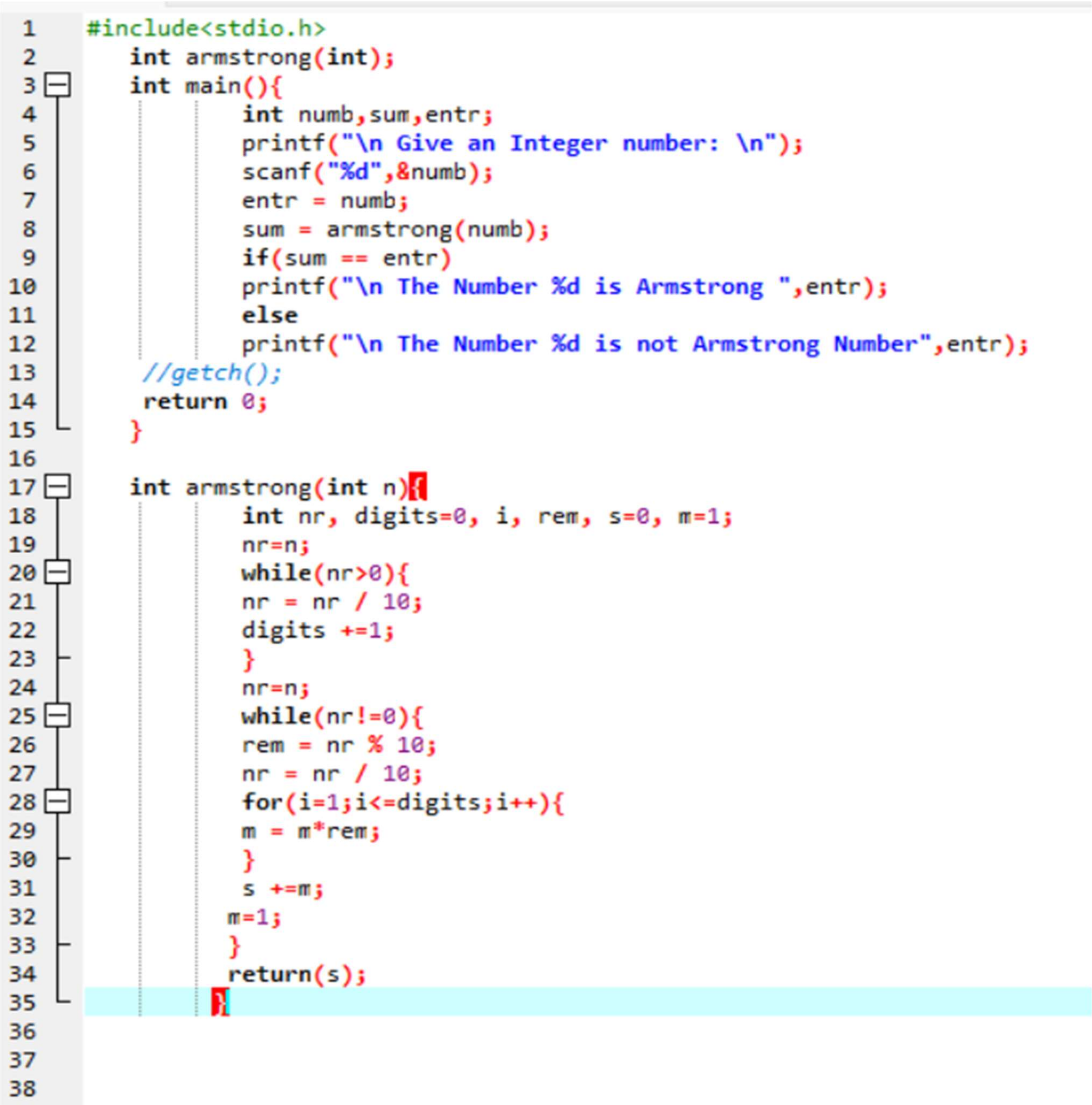
```c
#include <stdio.h>

int fact(int);

int main()
{
    int no,factorial;

    printf("Enter a number to calculate it's factorial\n");
    scanf("%d",&no);
    factorial=fact(no);
    printf("Factorial of the num(%d) = %d\n",no,factorial);
//printf("Factorial of the num(%d) = %d\n",no,fact(no));//another way of calling a function//comment above two lines if you want to use this
}

int fact(int n)
{
    int i,f=1;
    for(i=1;i<=n;i++)
    {
        f=f*i;
    }
    return f;
}




```

PRIME NOS USING FUNCTION



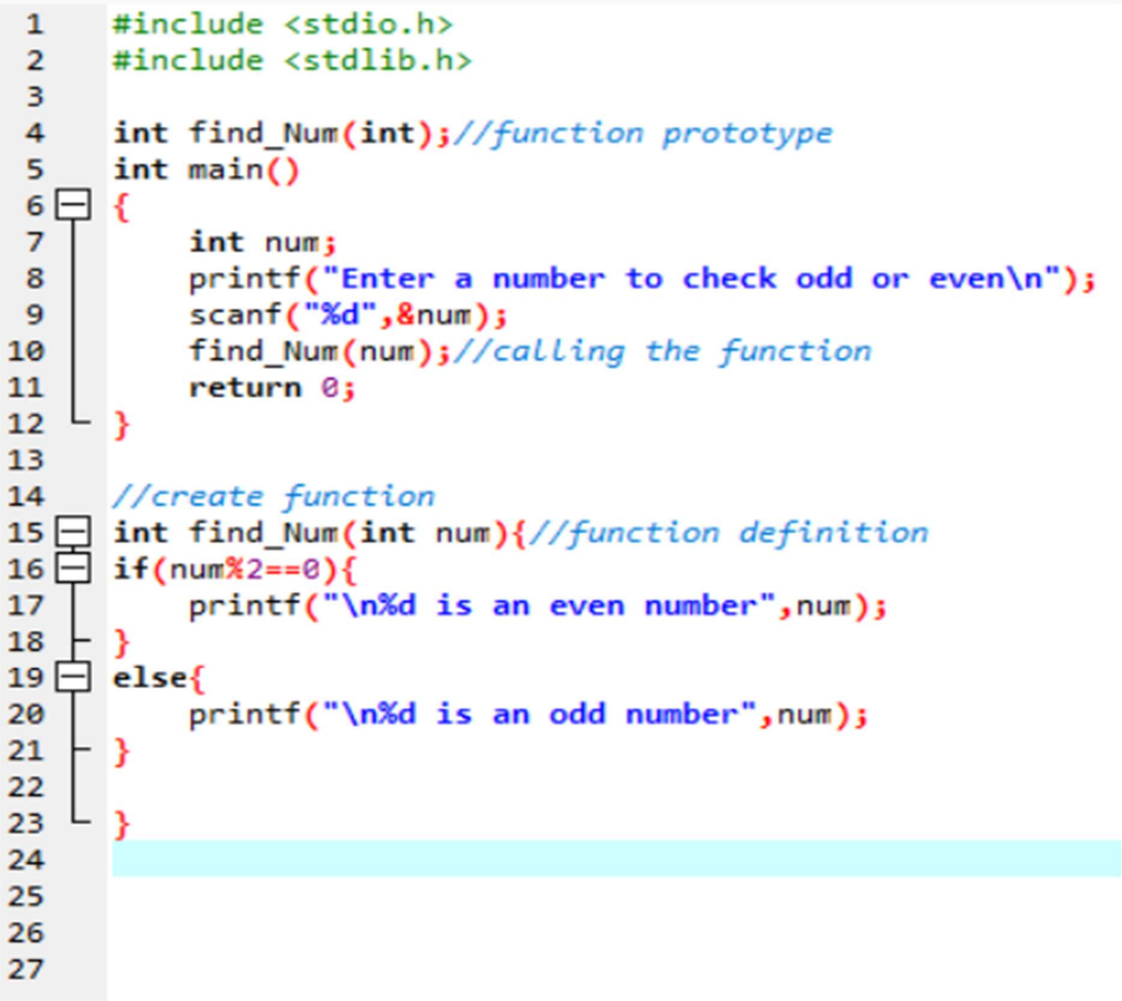
```c
#include <stdio.h>
void prime(int);
int main()
{
    int n;
    printf("Enter any number\n");
    scanf("%d",&n);
    prime(n);
}
void prime(int n)
{
    int i,c=0;
    for(i=1;i<= n;i++)
    {
        if (n%i==0)
        {
            c++;
        }
    }
    if(c==1)
    {
        printf("\n%d is neither prime nor number",n);
    }
    else if(c==2)
    {
        printf("\n%d is a Prime number",n);
    }
    else
    {
        printf("\n%d is not a Prime number",n);
    }
}
```

ARMSTRONG USING FUNCTION



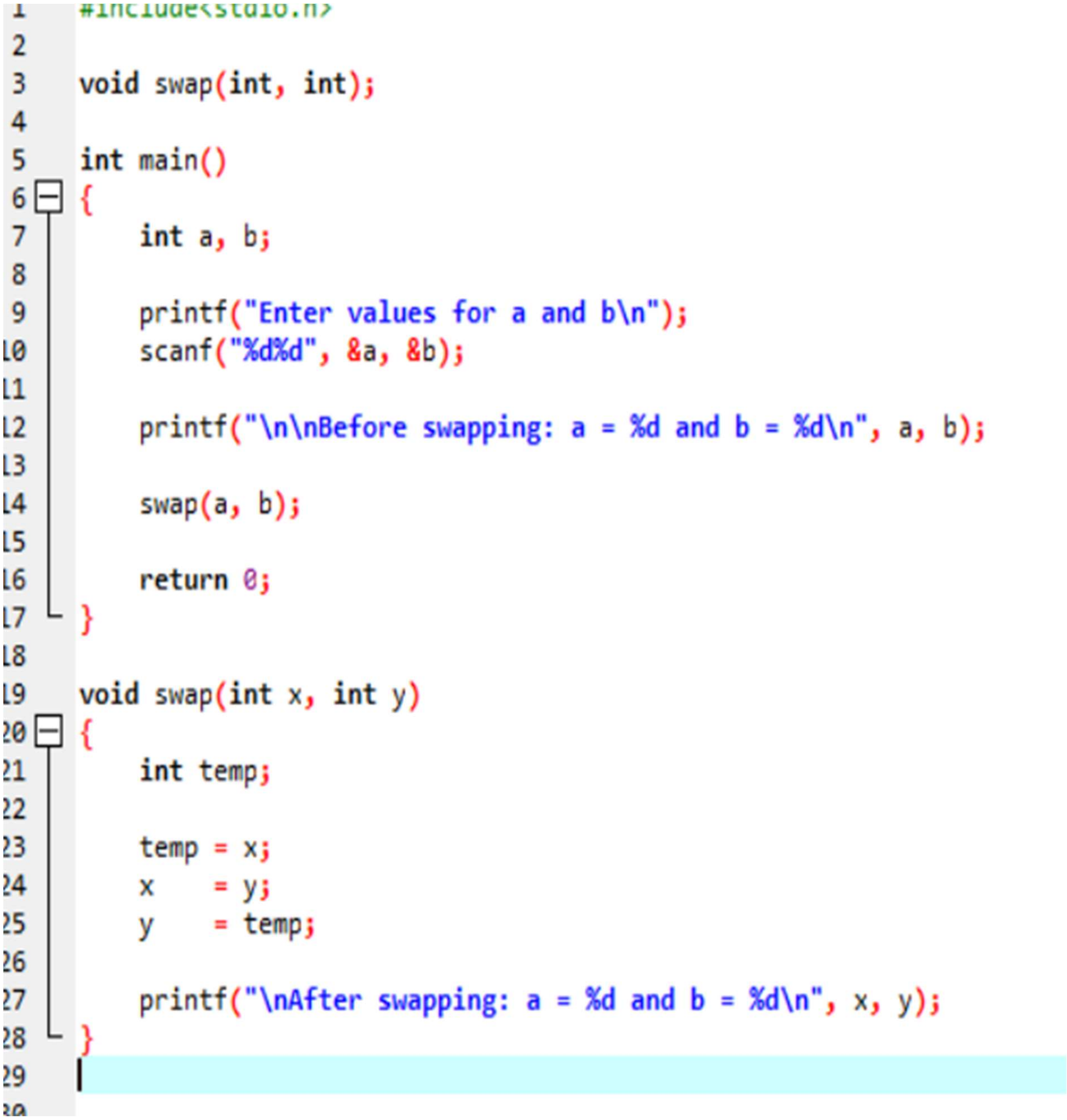
```
#include<stdio.h>
   int armstrong(int);
   int main(){
           int numb,sum,entr;
           printf("\n Give an Integer number: \n");
           scanf("%d",&numb);
           entr = numb;
           sum = armstrong(numb);
           if(sum == entr)
           printf("\n The Number %d is Armstrong ",entr);
           else
           printf("\n The Number %d is not Armstrong Number",entr);
    //getch();
    return 0;
   }

   int armstrong(int n){
           int nr, digits=0, i, rem, s=0, m=1;
           nr=n;
           while(nr>0){
           nr = nr / 10;
           digits +=1;
           }
           nr=n;
           while(nr!=0){
           rem = nr % 10;
           nr = nr / 10;
           for(i=1;i<=digits;i++){
           m = m*rem;
           }
           s +=m;
          m=1;
          }
          return(s);
         }
```

EVEN ODD USING FUNCTION



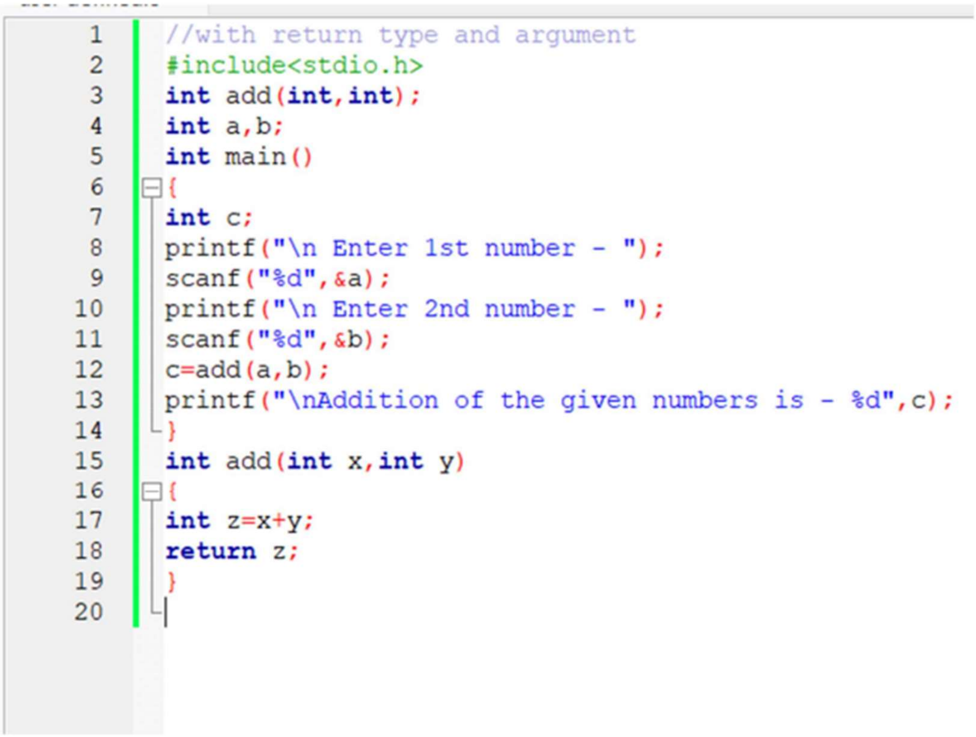
```c
#include <stdio.h>
#include <stdlib.h>

int find_Num(int);//function prototype
int main()
{
    int num;
    printf("Enter a number to check odd or even\n");
    scanf("%d",&num);
    find_Num(num);//calling the function
    return 0;
}

//create function
int find_Num(int num){//function definition
if(num%2==0){
    printf("\n%d is an even number",num);
}
else{
    printf("\n%d is an odd number",num);
}

}
```

SWAPPING USING FUNCTION

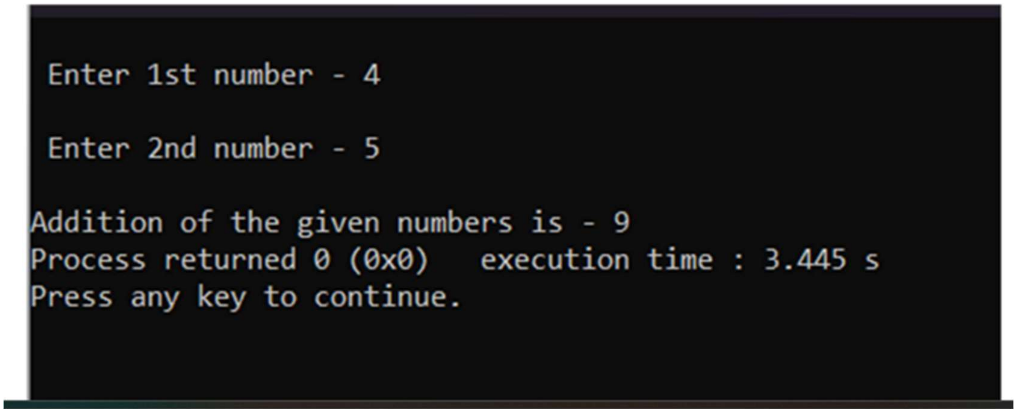


```
#include<stdio.h>

void swap(int, int);

int main()
{
    int a, b;

    printf("Enter values for a and b\n");
    scanf("%d%d", &a, &b);

    printf("\n\nBefore swapping: a = %d and b = %d\n", a, b);

    swap(a, b);

    return 0;
}

void swap(int x, int y)
{
    int temp;

    temp = x;
    x    = y;
    y    = temp;

    printf("\nAfter swapping: a = %d and b = %d\n", x, y);
}
```

ADDITION

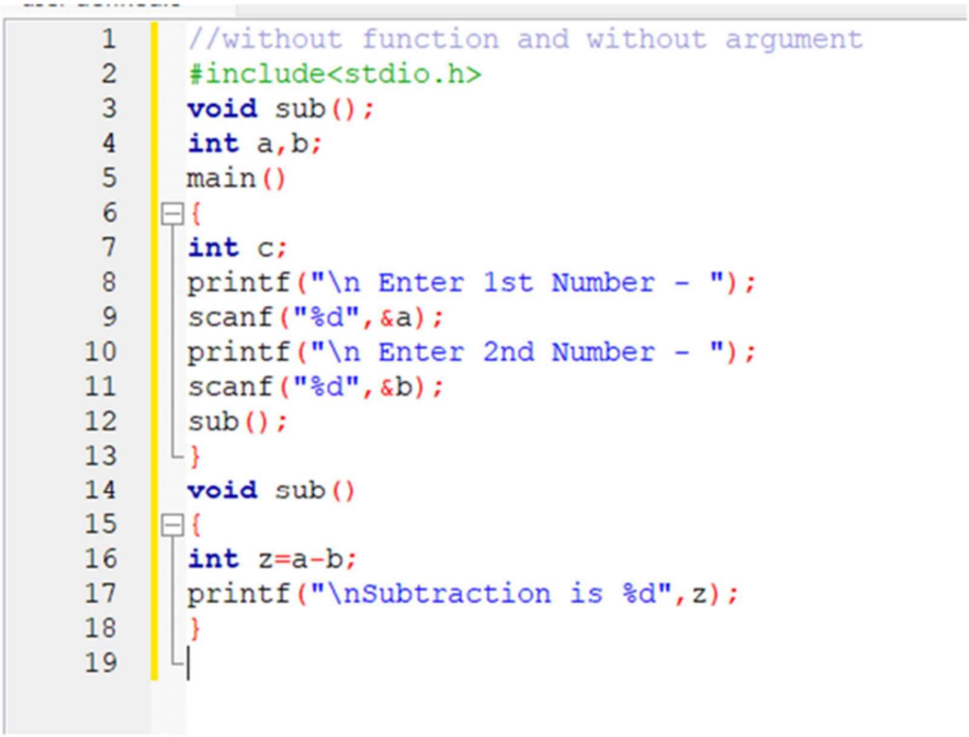


```c
//with return type and argument
#include<stdio.h>
int add(int,int);
int a,b;
int main()
{
int c;
printf("\n Enter 1st number - ");
scanf("%d",&a);
printf("\n Enter 2nd number - ");
scanf("%d",&b);
c=add(a,b);
printf("\nAddition of the given numbers is - %d",c);
}
int add(int x,int y)
{
int z=x+y;
return z;
}
```

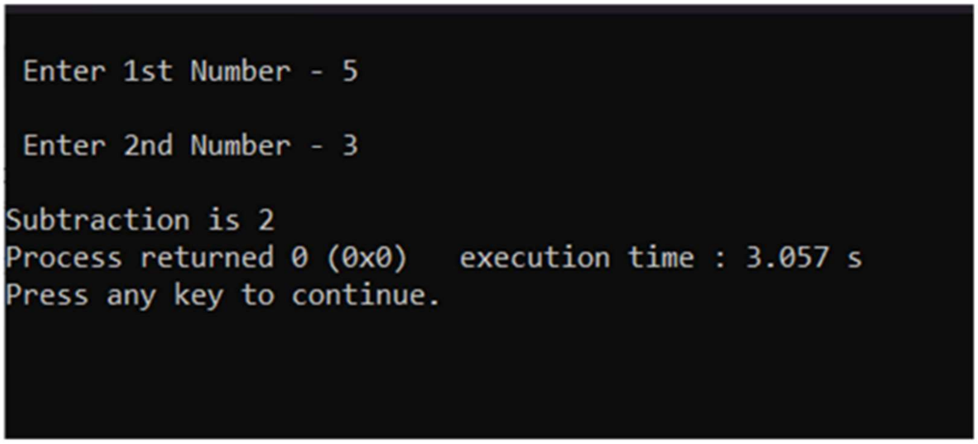


```
 Enter 1st number - 4

 Enter 2nd number - 5

Addition of the given numbers is - 9
Process returned 0 (0x0)   execution time : 3.445 s
Press any key to continue.
```

SUBTRACTION

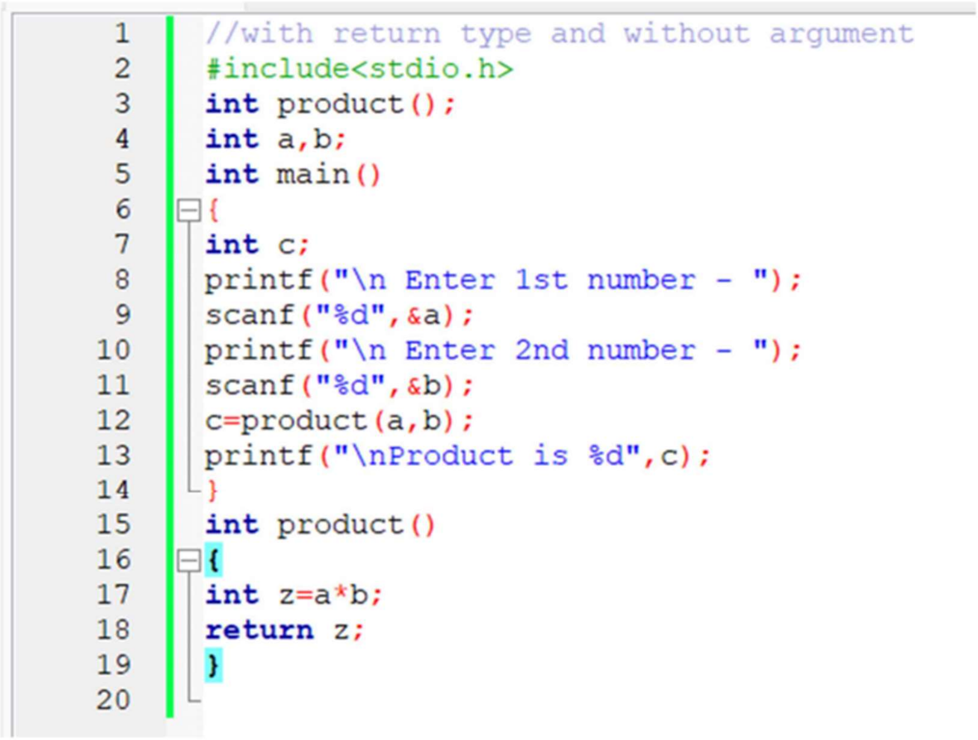


```c
//without function and without argument
#include<stdio.h>
void sub();
int a,b;
main()
{
int c;
printf("\n Enter 1st Number - ");
scanf("%d",&a);
printf("\n Enter 2nd Number - ");
scanf("%d",&b);
sub();
}
void sub()
{
int z=a-b;
printf("\nSubtraction is %d",z);
}
```

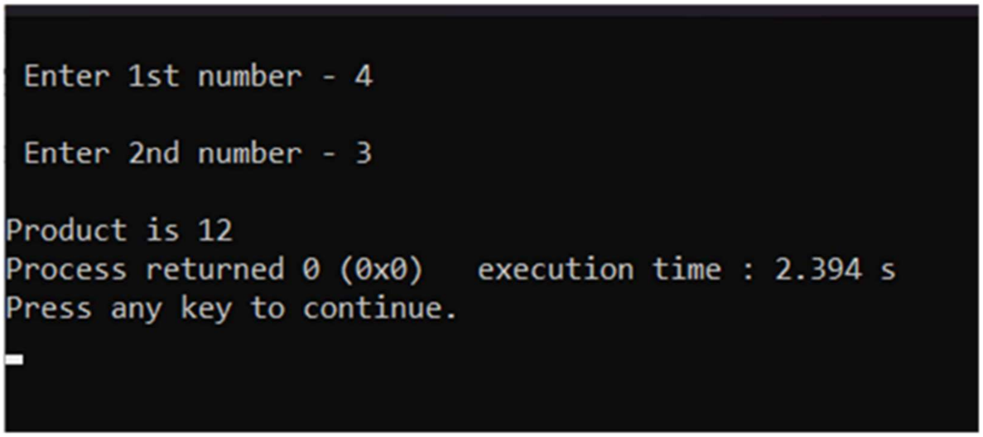


```
 Enter 1st Number - 5

 Enter 2nd Number - 3

Subtraction is 2
Process returned 0 (0x0)   execution time : 3.057 s
Press any key to continue.
```

MULTIPLICATION

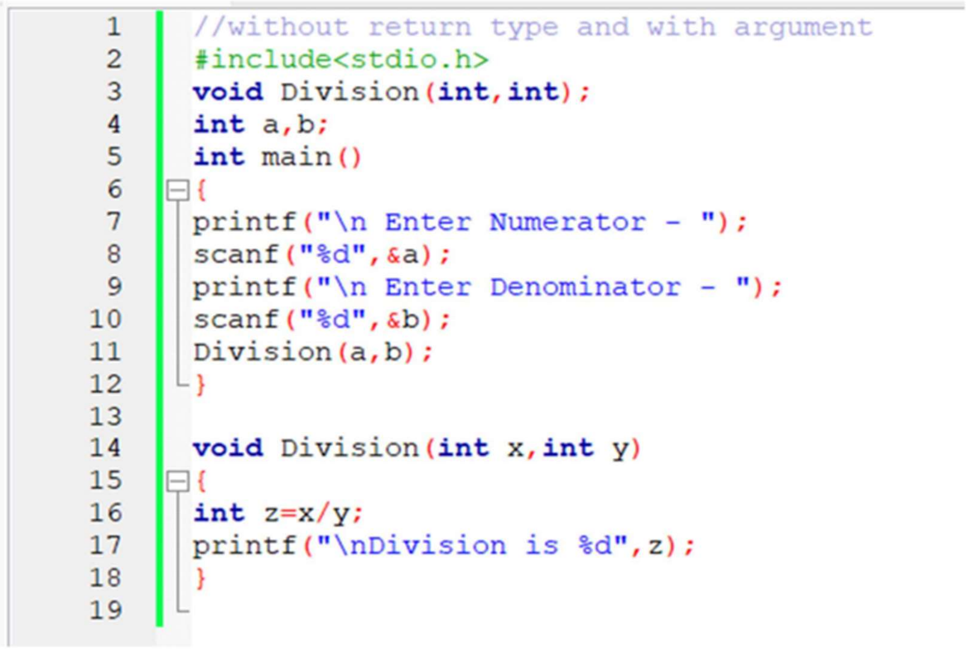


```c
//with return type and without argument
#include<stdio.h>
int product();
int a,b;
int main()
{
int c;
printf("\n Enter 1st number - ");
scanf("%d",&a);
printf("\n Enter 2nd number - ");
scanf("%d",&b);
c=product(a,b);
printf("\nProduct is %d",c);
}
int product()
{
int z=a*b;
return z;
}
```

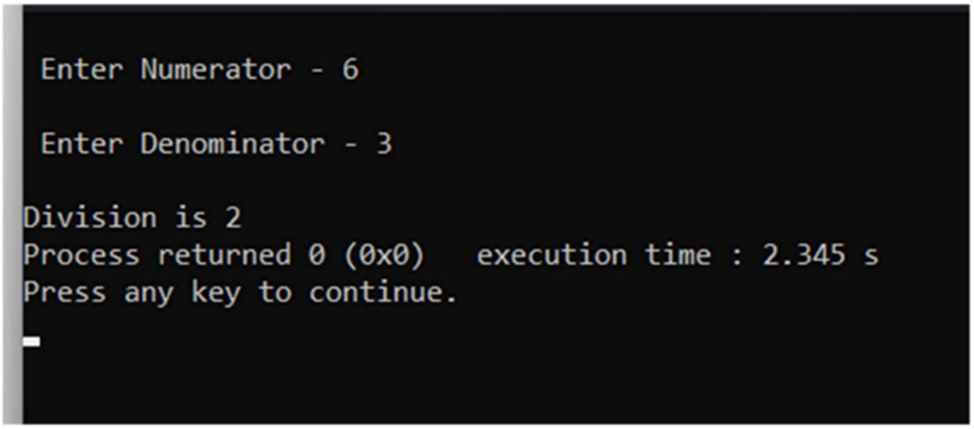


```
 Enter 1st number - 4

 Enter 2nd number - 3

Product is 12
Process returned 0 (0x0)   execution time : 2.394 s
Press any key to continue.
```

DIVISION

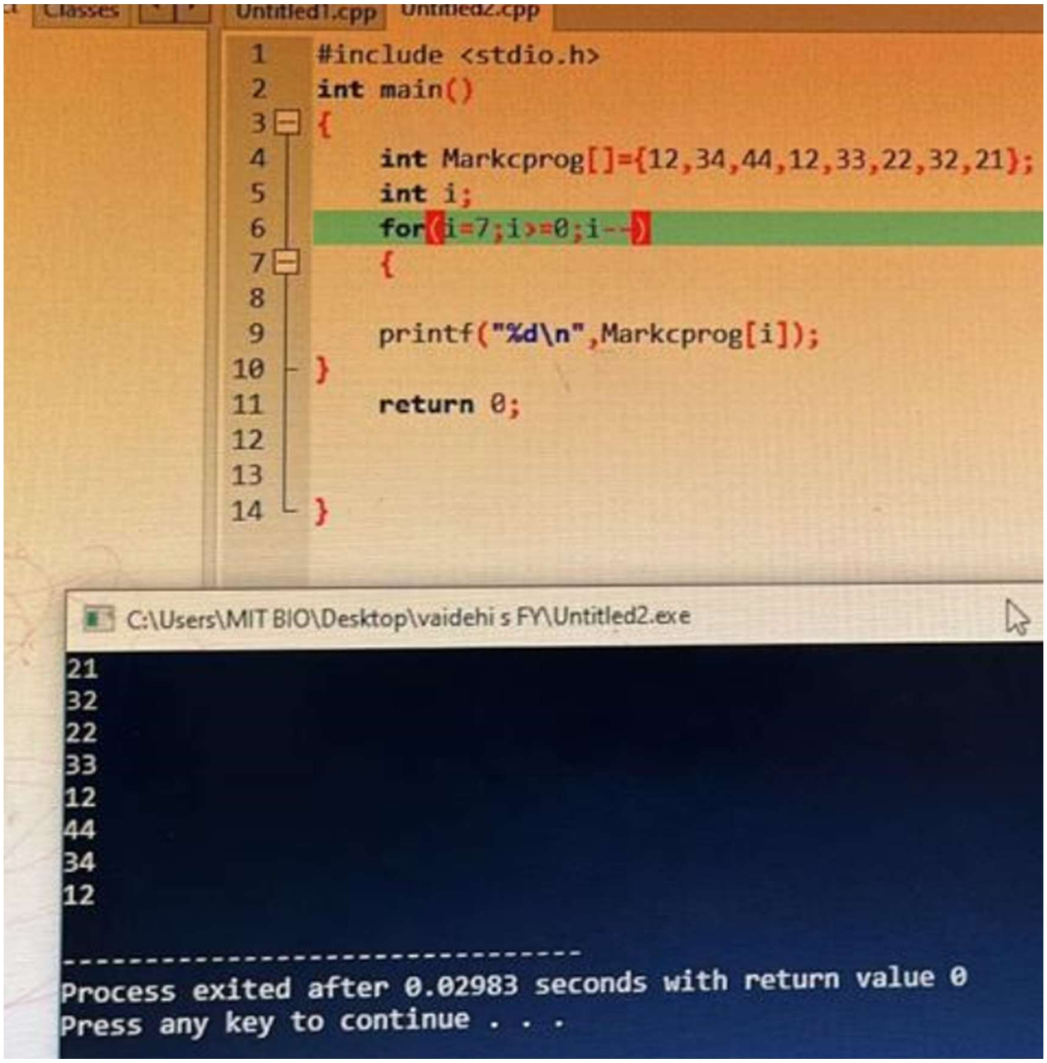


```c
//without return type and with argument
#include<stdio.h>
void Division(int,int);
int a,b;
int main()
{
printf("\n Enter Numerator - ");
scanf("%d",&a);
printf("\n Enter Denominator - ");
scanf("%d",&b);
Division(a,b);
}

void Division(int x,int y)
{
int z=x/y;
printf("\nDivision is %d",z);
}
```



```
 Enter Numerator - 6

 Enter Denominator - 3

Division is 2
Process returned 0 (0x0)   execution time : 2.345 s
Press any key to continue.
```

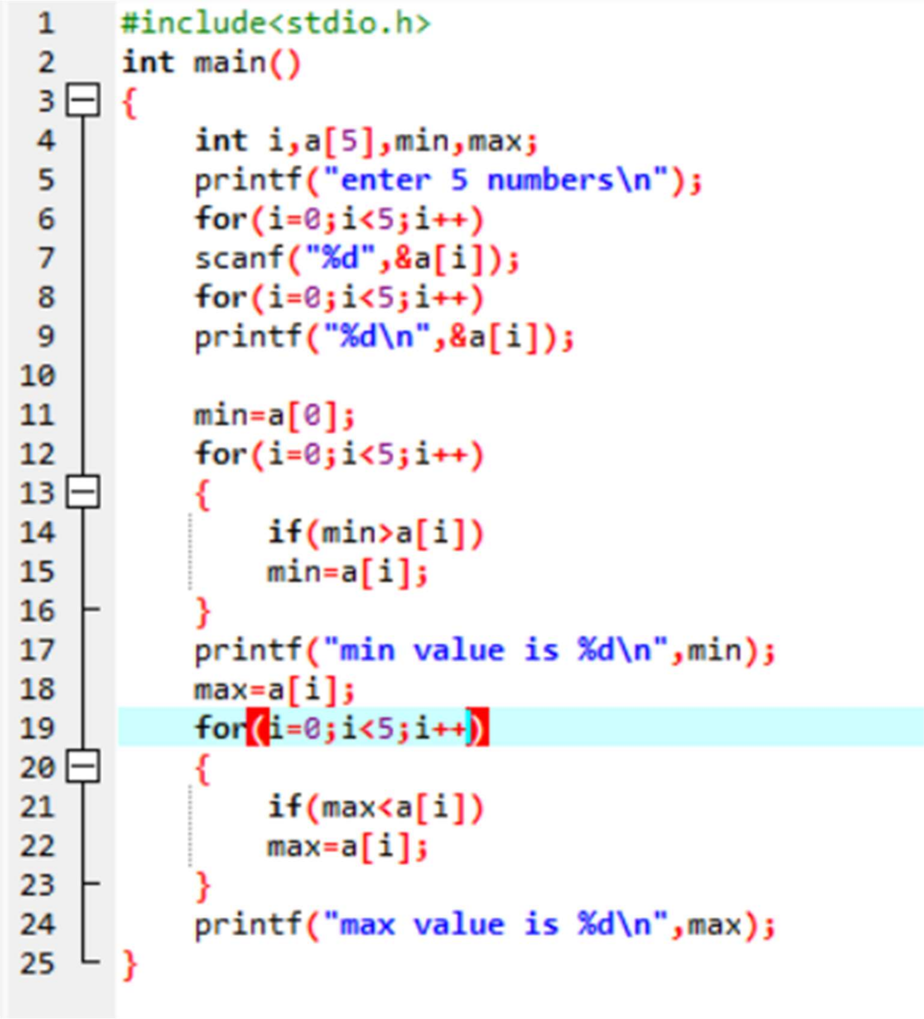
REVERSE ARRAY



```
#include <stdio.h>
int main()
{
	int Markcprog[]={12,34,44,12,33,22,32,21};
	int i;
	for(i=7;i>=0;i--)
	{

	printf("%d\n",Markcprog[i]);
}
	return 0;


}
```

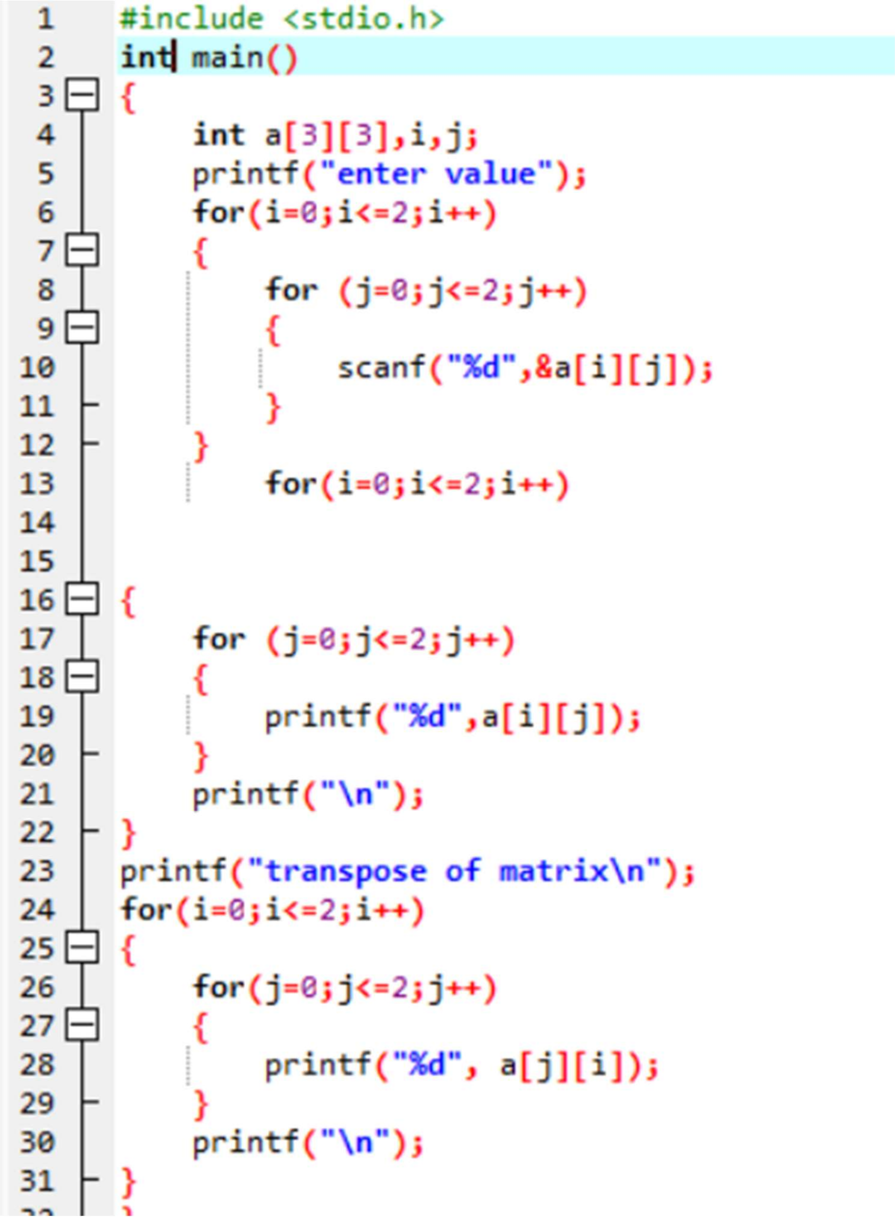
```
21
32
22
33
12
44
34
12

--------------------------------
Process exited after 0.02983 seconds with return value 0
Press any key to continue . . .
```

ARRAY MIN MAX



```c
#include<stdio.h>
int main()
{
    int i,a[5],min,max;
    printf("enter 5 numbers\n");
    for(i=0;i<5;i++)
    scanf("%d",&a[i]);
    for(i=0;i<5;i++)
    printf("%d\n",&a[i]);

    min=a[0];
    for(i=0;i<5;i++)
    {
        if(min>a[i])
        min=a[i];
    }
    printf("min value is %d\n",min);
    max=a[i];
    for(i=0;i<5;i++)
    {
        if(max<a[i])
        max=a[i];
    }
    printf("max value is %d\n",max);
}
```

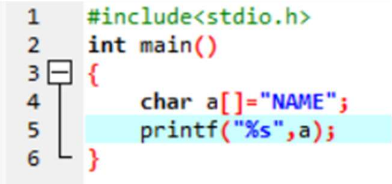
TRANSPOSE OF MATRIX



```c
#include <stdio.h>
int main()
{
    int a[3][3],i,j;
    printf("enter value");
    for(i=0;i<=2;i++)
    {
        for (j=0;j<=2;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
        for(i=0;i<=2;i++)


{
    for (j=0;j<=2;j++)
    {
        printf("%d",a[i][j]);
    }
    printf("\n");
}
printf("transpose of matrix\n");
for(i=0;i<=2;i++)
{
    for(j=0;j<=2;j++)
    {
        printf("%d", a[j][i]);
    }
    printf("\n");
}
}
```

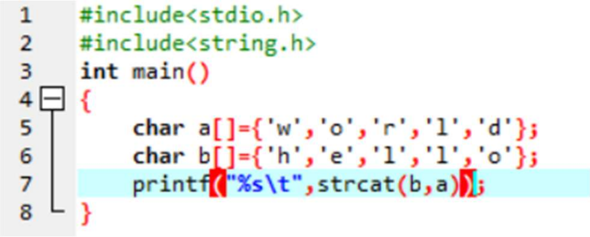
STRINGS

1.



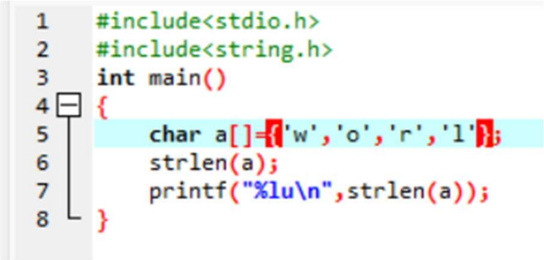
```c
#include<stdio.h>
int main()
{
    char a[]="NAME";
    printf("%s",a);
}
```

2.



```c
#include<stdio.h>
#include<string.h>
int main()
{
    char a[]={'w','o','r','l','d'};
    char b[]={'h','e','l','l','o'};
    printf("%s\t",strcat(b,a));
}
```

3.



```c
#include<stdio.h>
#include<string.h>
int main()
{
    char a[]={'w','o','r','l'};
    strlen(a);
    printf("%lu\n",strlen(a));
}
```