

## 2).**armstrong number are not:**

```
#include<stdio.h>

#include<math.h>

int armstrong(int n)
{
    int d,temp;

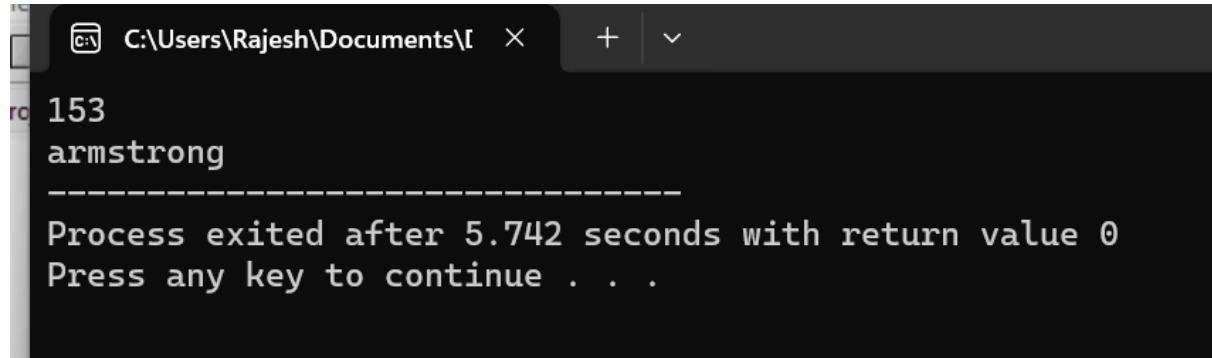
    if(n==0)
    {
        return 0;
    }

    else
    {
        d=n%10;
        return (pow(d,3)+armstrong(n/10));
    }
}

int main()
{
    int n,temp;
    scanf("%d",&n);
    if(armstrong(n)==n)
    {
        printf("armstrong");
    }
    else
    {
        printf(" not armstrong");
    }
}
```

```
        return 0;
    }
}
```

Output:



```
C:\Users\Rajesh\Documents\I  ×  +  ∨
153
armstrong
-----
Process exited after 5.742 seconds with return value 0
Press any key to continue . . .
```

### 3).gcd using recursion:

```
#include<stdio.h>

int gcd(int a,int b)
{
    while(a!=b)
    {
        if(a>b)
        {
            return gcd(a-b,b);
        }
        else
        {
            return gcd(a,b-a);
        }
    }
}

int main()
{
    int a,b;

    printf("enteert the number:");

    scanf("%d %d",&a,&b);
```

```

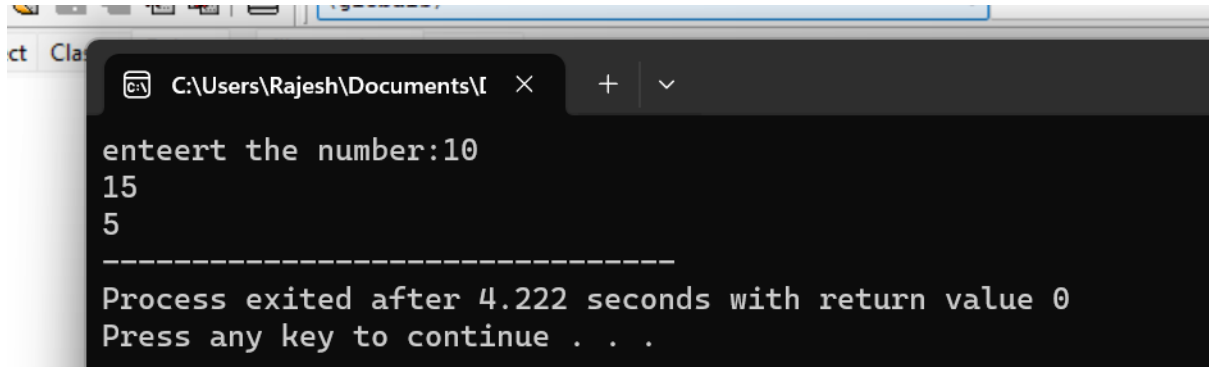
    printf("%d",gcd(a,b));

    return 0;

}

```

Output:



```

C:\Users\Rajesh\Documents\I
enteert the number:10
15
5
-----
Process exited after 4.222 seconds with return value 0
Press any key to continue . . .

```

#### 4)largest number in array:

```

#include<stdio.h>

int main()
{
    int a[10],max,n,i;

    printf("enter the size of array:");

    scanf("%d",&n);

    printf("enter the array elements:");

    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }

    max=a[0];

    for(i=0;i<n;i++)
    {
        if(a[i]>max)
        {
            max=a[i];
        }
    }

    printf("%d",max);

}

```

Output:

```
enter the size of array:5
enter the array elements:12
15
12
18
6456
6456
-----
Process exited after 11.09 seconds with return value 0
Press any key to continue . . . |
```

## 5).factorial number:

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

```
int fact(int n)
```

```
{
```

```
    if(n==0)
```

```
    {
```

```
        return 1;
```

```
    }
```

```
    else
```

```
    {
```

```
        return n*(fact(n-1));
```

```
    }
```

```
}
```

```
int main()
```

```
{
```

```
    int n;
```

```
    printf("enter the number:");
```

```
    scanf("%d",&n);
```

```
//    fact(n);
```

```
    printf("%d",fact(n))
```

output:

```
ct
enter the number:5
120
-----
Process exited after 2.36 seconds with return value 0
Press any key to continue . . . |
```

6) **prime number are not:**

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

```
int main()
```

```
{
```

```
    int n,i,flag;
```

```
    printf("enter the number:");
```

```
    scanf("%d",&n);
```

```
    flag=0;
```

```
    for(i=2;i<=n/2;i++)
```

```
    {
```

```
        if(n%i==0)
```

```
        {
```

```
            //    printf("%d",i);
```

```
            flag=1;
```

```
            break;
```

```
        }
```

```
    }
```

```
    if(flag==1)
```

```
    {
```

```
        printf("not prime number");
```

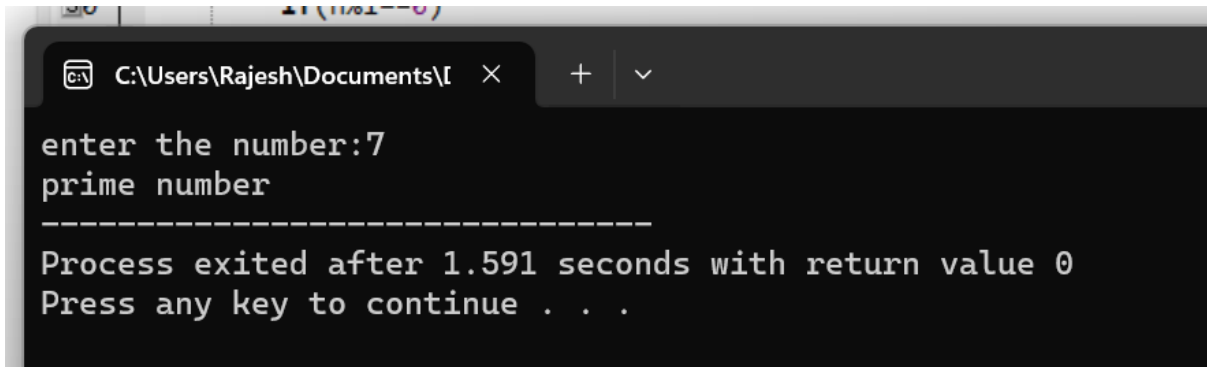
```
    }
```

```

        else
        {
            printf("prime number");
        }
    }
}

```

Output:



```

C:\Users\Rajesh\Documents\I
enter the number:7
prime number
-----
Process exited after 1.591 seconds with return value 0
Press any key to continue . . .

```

## 7)selection sorting:

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

```
int selection(int n,int a[])
```

```
{
```

```
    int i,j,small,temp;
```

```
    for(i=0;i<n;i++)
```

```
    {
```

```
        small=i;
```

```
        for(j=i+1;j<n;j++)
```

```
        {
```

```
            if(a[j]<a[small])
```

```
            {
```

```
                small=j;
```

```
            temp=a[small];
```

```
            a[small]=a[i];
```

```
            a[i]=temp;
```

```

        }
    }
}

int main()
{
    int n,a[10];
    int i;
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    selection(n,a);
    printf("after the sorting");
    for(i=0;i<n;i++)
    {
        printf("%d",a[i]);
    }
}

```

Output:

```

5
12
31
25
8
24
after the sorting824122531
-----

```

Process exited after 7.399 seconds with re

## 8) **bubble sort:**

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

```
int bubble(int a[],int n)
```

```
{
```

```
    int i,j;
```

```
    for(i=0;i<n-1;i++)
```

```
    {
```

```
        for(j=0;j<n-i-1;j++)
```

```
        {
```

```
            if(a[j]>a[j+1])
```

```
            {
```

```
                int temp=a[j];
```

```
                a[j]=a[j+1];
```

```
                a[j+1]=temp;
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

```
int main()
```

```
{
```

```
    int a[10];
```

```
    int i,n;
```

```
    scanf("%d",&n);
```

```
    for(i=0;i<n;i++)
```

```
    {
```

```
        scanf("%d",&a[i]);
```

```
    }
```

```
    bubble(a,n);
```

```
    for(i=0;i<n;i++)
```

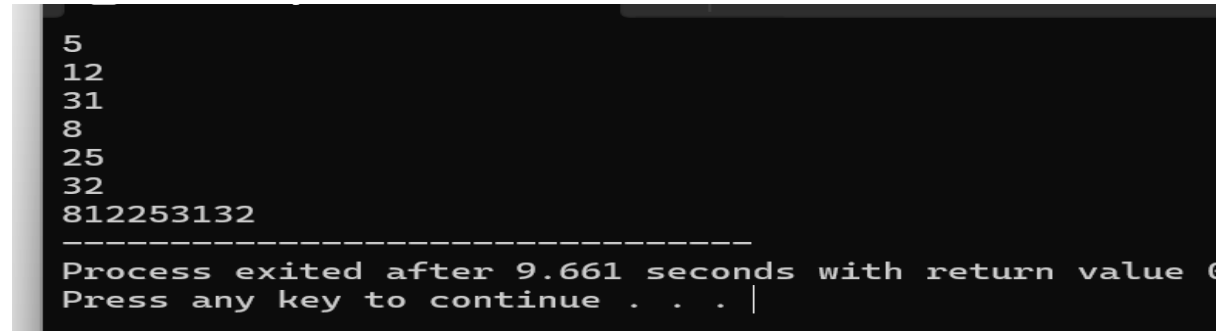


```

    {
        printf("%d",a[i]);
    }
}

```

Output:



```

5
12
31
8
25
32
812253132
-----
Process exited after 9.661 seconds with return value 0
Press any key to continue . . . |

```

## 9) matrix multiplication

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

```
int main()
```

```

{
    int a[10][10],b[10][10],mu[10][10];
    int i,j,k,r,c;
    printf("enter the number row:");
    scanf("%d",&r);
    printf("enter the number columns:");
    scanf("%d",&c);
    printf("enter the first matrix:");
    for(i=0;i<c;i++)
    {
        for(j=0;j<r;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
}

```

```

printf("enter the second matrix:");
for(i=0;i<c;i++)
{
    for(j=0;j<r;j++)
    {
        scanf("%d",&b[i][j]);
    }
}
for(i=0;i<c;i++)
{
    for(j=0;j<r;j++)
    {
        mu[i][j]=0;
        for(k=0;k<r;k++)
        {
            mu[i][j]+=a[i][k]*b[k][j];
        }
    }
}
for(i=0;i<c;i++)
{
    for(j=0;j<r;j++)
    {
        printf("%d",mu[i][j]);
    }
}
}

```

Output:

```
C:\Users\Rajesh\Documents\l x + v
enter the number row:2
enter the number columns:2
enter the first matrix:1
1
1
1
enter the second matrix:2
2
2
2
4444
-----
Process exited after 9.595 seconds with return value 0
Press any key to continue . . .
```

10)string palindrome are not:

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

```
#include<string.h>
```

```
int main()
```

```
{
```

```
    char s[10];
```

```
    int l,i,flag=0;
```

```
    scanf("%s",s);
```

```
    l=strlen(s);
```

```
    for(i=0;i<l/2;i++)
```

```
    {
```

```
        if(s[i]!=s[l-i-1])
```

```
        {
```

```
            flag=1;
```

```
            break;
```

```
        }
```

```
    }
```

```
    if(flag==1)
```

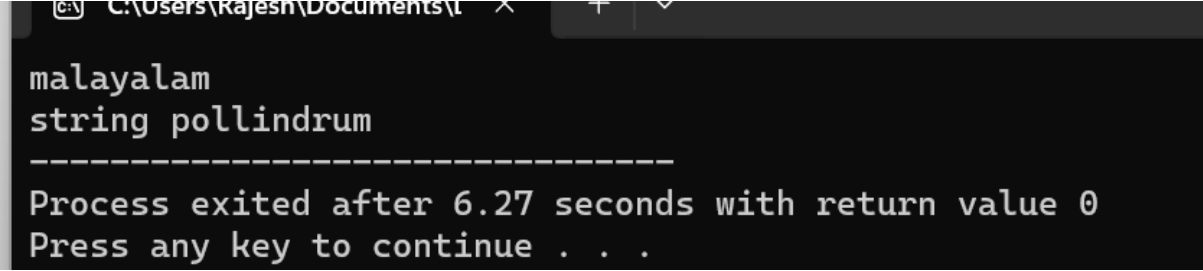
```
    {
```

```
        printf("string not pollindrum");
```

```
    }
```

```
    else
    {
        printf("string pollindrum");
    }
}
```

Output:



```
malayalam
string pollindrum
-----
Process exited after 6.27 seconds with return value 0
Press any key to continue . . .
```