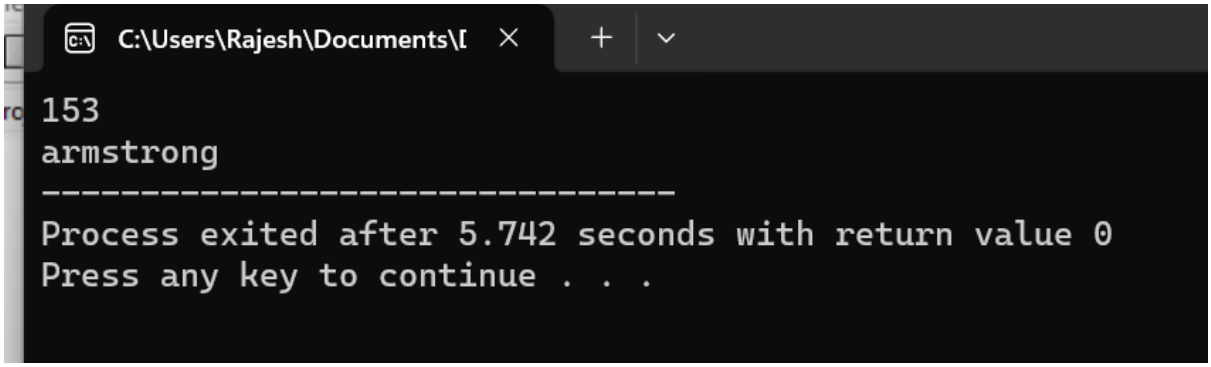
2).armstrong number are not:

```
#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:

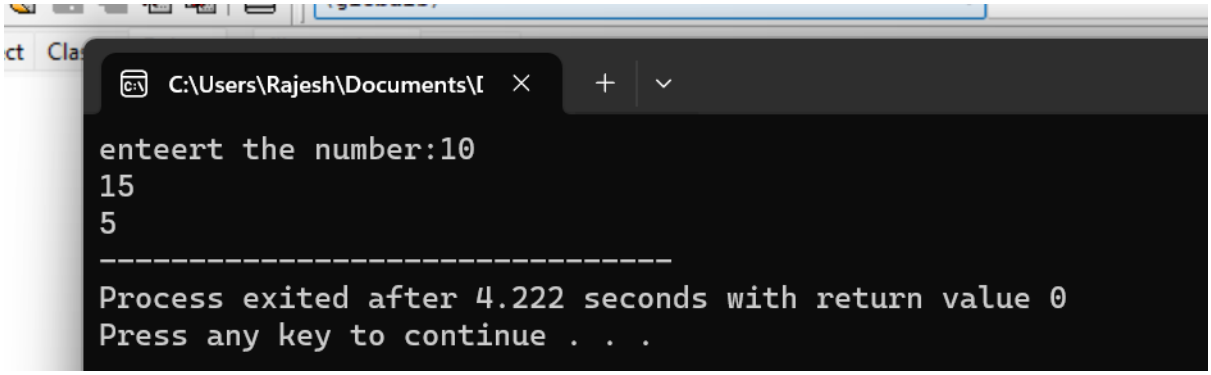


3).gcd using recursion:

```
#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;
}
```

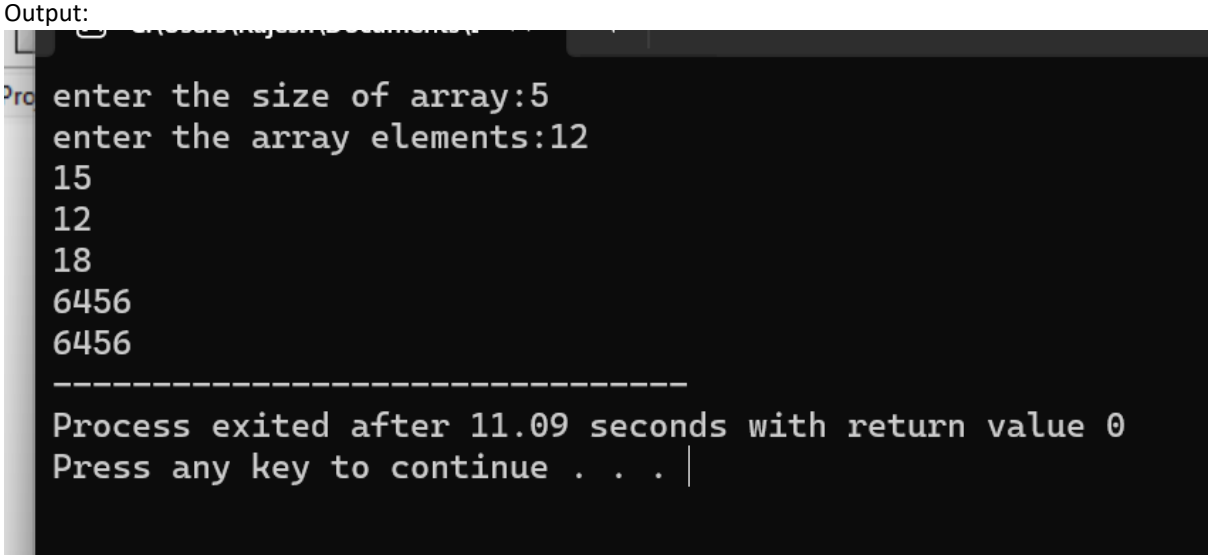
Output:



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);
}
```

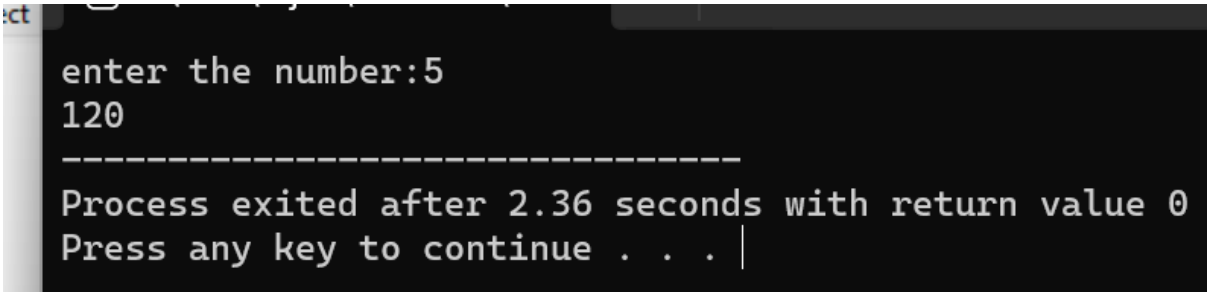
Output:



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:

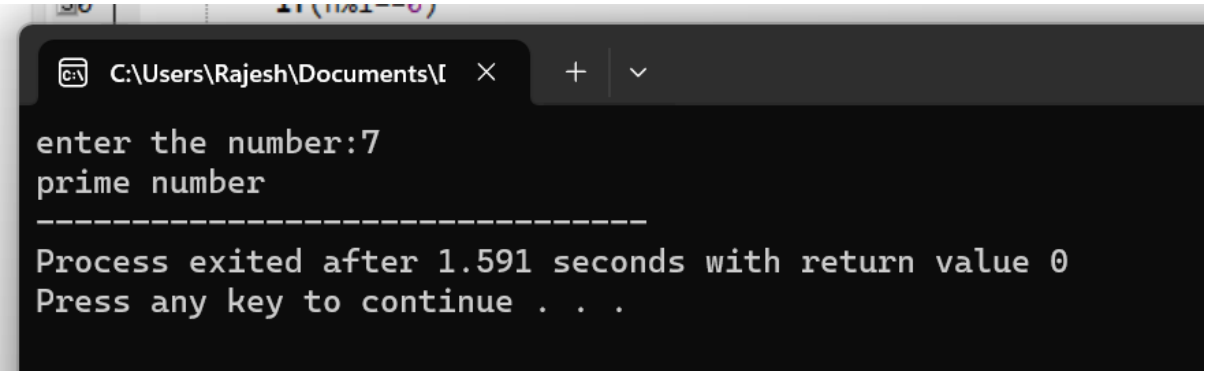


6) **prime number are not:**

```
#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:



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:

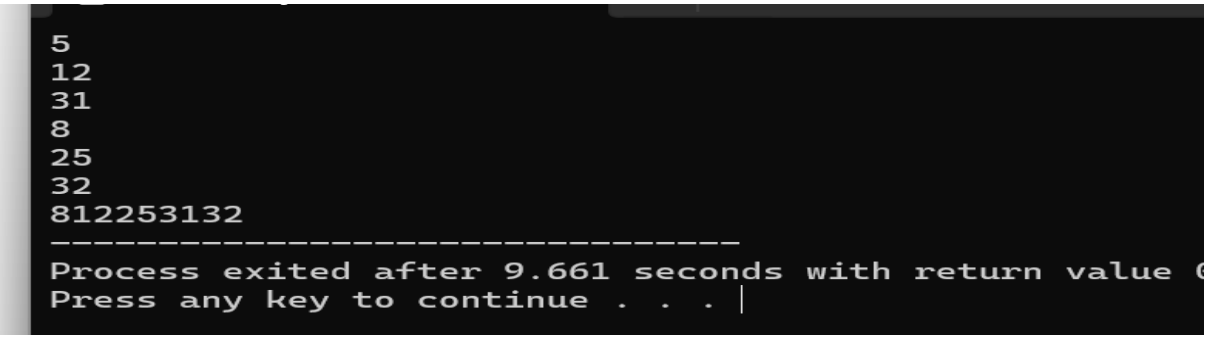
```
5
12
31
25
8
24
after the sorting824122531
--------------------------------
Process exited after 7.399 seconds with re
```

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:

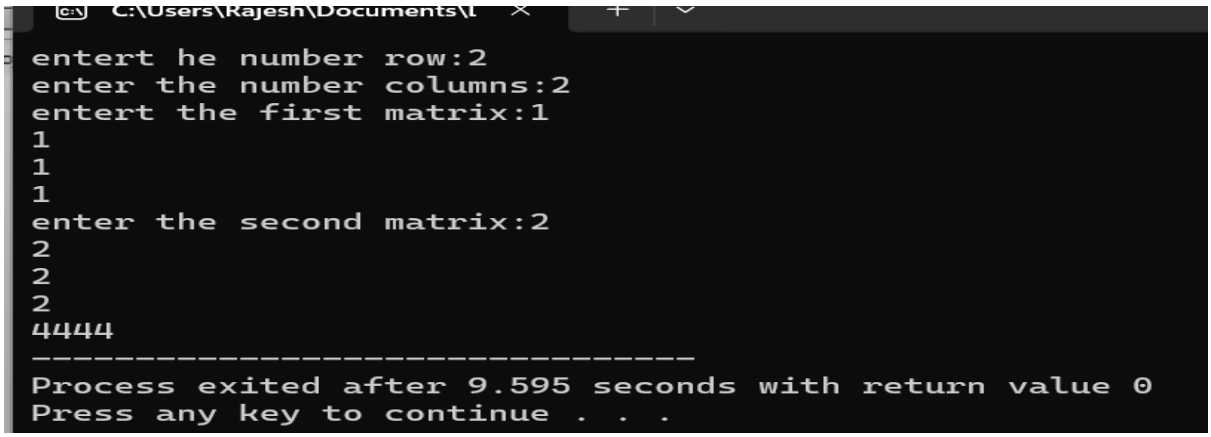


9) **matrix multiplication**

```
#include<stdio.h>
int main()
{
        int a[10][10],b[10][10],mu[10][10];
        int i,j,k,r,c;
        printf("entert he number row:");
        scanf("%d",&r);
        printf("enter the number columns:");
        scanf("%d",&c);
        printf("entert 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:



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: