- **To print various pyramids pattern.**

>>> Syntax :

```
#include<stdio.h>


void pattern1()
{
	for(int i=0;i<6;i++)
	{
		for(int j=0;j<i;j++)
			{
				printf("* ");
			}
		printf("\n");
	}
}

void pattern2()
{
	for(int i=0;i<6;i++)
	{
		for(int j=0;j<i;j++)
			{
				printf("%d ",i);
			}
		printf("\n");
	}
}

void pattern3()
{
	for(int i=0;i<7;i++)
	{
		for(int j=1;j<i;j++)
			{
				printf("%d ",j);
			}
		printf("\n");
	}
}

void pattern4()
{
	int count=1;
	for(int i=0;i<6;i++)
	{
		for(int j=0;j<i;j++)
			{
				printf("%d ",count);
				count++;
			}
		printf("\n");
```

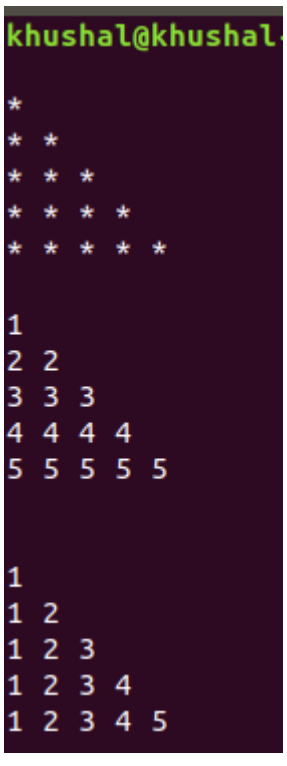
```
	}
}

void pattern5()
{
	int count=1;
	for(int i=0;i<6;i++)
	{
		for(int j=0;j<i;j++)
			{
				count++;
				printf("%d ",count);
			}
		count--;
		printf("\n");
	}
}

void pattern6()
{
	for(int i=0;i<6;i++)
	{
		for(int j=0;j<i;j++)
			{
				printf("%d ",i+j);
			}
		printf("\n");
	}
}

void pattern7()
{
	for(int i=0;i<6;i++)
	{
		for(int j=0;j<i;j++)
		{
			if((i+j)%2==0)
				printf("0 ");
			else
				printf("1 ");
		}
		printf("\n");
	}
}

void pattern8()
{
	for(int i=5;i>=0;i--)
	{
		for(int j=5;j>i;j--)
			{
				printf("%d ",j);
```

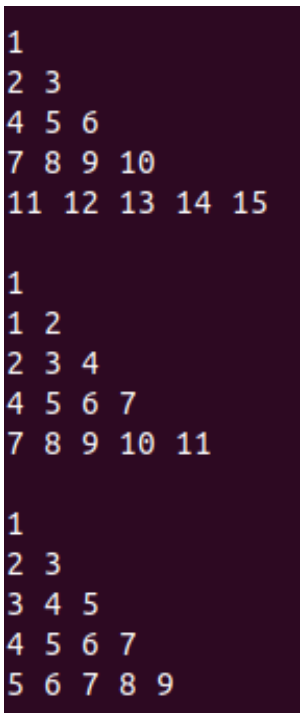
```
                }
            printf("\n");
        }
}

void pattern9()
{
        for(int i=5;i>0;i--)
        {
            for(int j=6;j>i;j--)
                {
                    printf("%d ",i);
                }
            printf("\n");
        }
}

void pattern10()
{
        for(int i=5;i>0;i--)
        {
            for(int j=0;j<i;j++)
                {
                    printf("* ");
                }
            printf("\n");
        }
}

void pattern11()
{
        for(int i=5;i>0;i--)
        {
            for(int j=0;j<i;j++)
                {
                    printf("%d ",i);
                }
            printf("\n");
        }
}

void pattern12()
{
        for(int i=5;i>0;i--)
        {
            for(int j=1;j<=i;j++)
                {
                    printf("%d ",j);
                }
            printf("\n");
        }
}
```

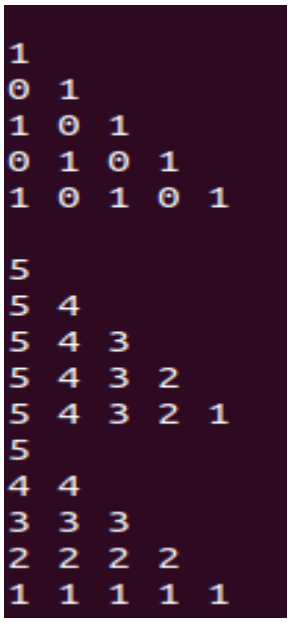
```
void pattern13()
{
	for(int i=5;i>0;i--)
	{
		for(int j=1;j<=i;j++)
			{
				printf("%d ",i-j+1);
			}
		printf("\n");
	}
}

void pattern14()
{
	for(int i=5;i>0;i--)
	{
		int count=5;
		for(int j=1;j<=i;j++)
			{
				printf("%d ",count);
				count--;
			}
		printf("\n");
	}
}

void pattern15()
{
	for(int i=1;i<=5;i++)
	{
		for(int j=5;j>=i;j--)
			{
				printf("%d ",i);
			}
		printf("\n");
	}
}

void pattern16()
{
	for(int i=1;i<=5;i++)
	{
		for(int j=1;j<=5-i;j++)
			printf(" ");
		for(int j=1;j<=i;j++)
			printf("*");
		printf("\n") ;
	}
}

void pattern17()
{
```

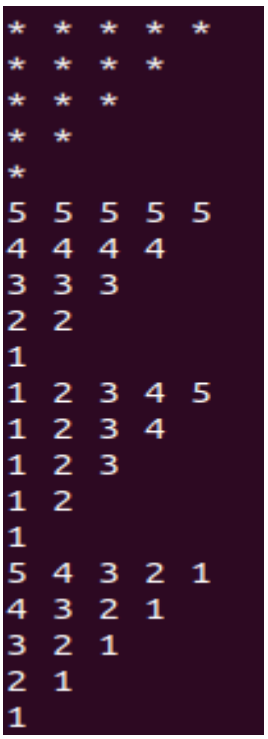
```
        for(int i=1;i<=5;i++)
        {
                for(int j=1;j<=5-i;j++)
                        printf(" ");
                for(int j=1;j<=i;j++)
                        printf("* ");
                printf("\n") ;
        }
}

int main(void)
{
        pattern1();
        pattern2();
        pattern3();
        pattern4();
        pattern5();
        pattern6();
        pattern7();
        pattern8();
        pattern9();
        pattern10();
        pattern11();
        pattern12();
        pattern13();
        pattern14();
        pattern15();
        pattern16();
        pattern17();
        return 0;
}
```

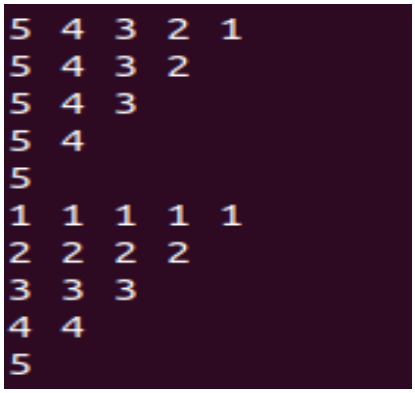
>>> **Output :**

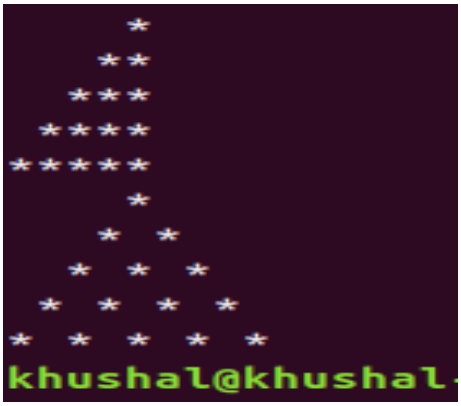












- **Program for various functions : Factorial, Prime ,Amstrong.**

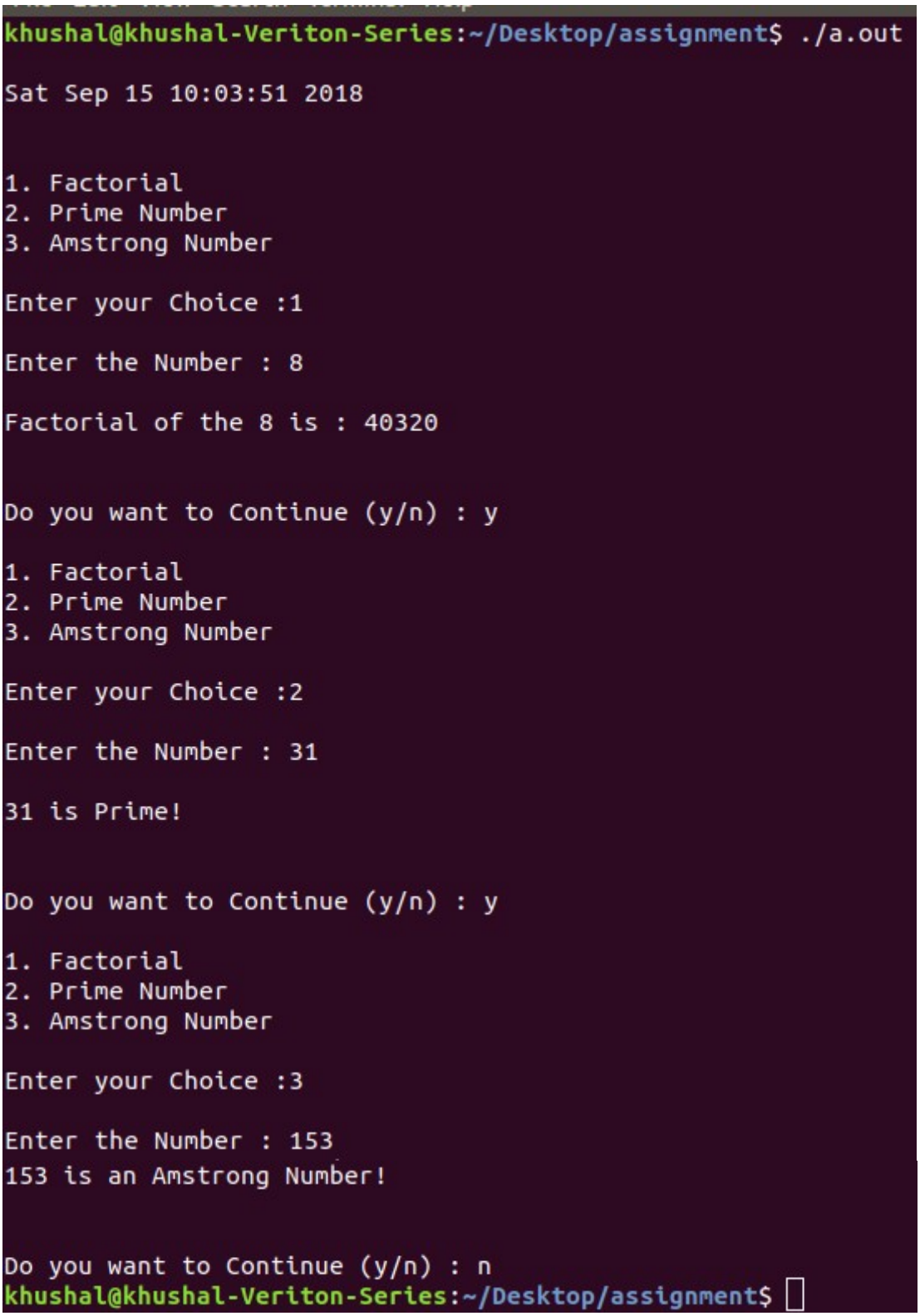
>>> Syntax :

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

void factorial(int);
void prime(int);
void amstrong(int);
void timestamp();

int main(void)
{
	int num,ch,y;
	char str='Y';
	timestamp();
	while(str=='Y' || str=='y')
	{
		printf("\n1. Factorial \n2. Prime Number \n3. Amstrong Number \n\nEnter your
Choice :");
		scanf("%d",&ch);
		printf("\nEnter the Number : ");
		scanf("%d",&num);
		switch (ch)
		{
			case 1: factorial(num);break;
			case 2: prime(num);break;
			case 3: amstrong(num);break;
		}
		printf("\nDo you want to Continue (y/n) : ");
		scanf("%s",&str);
	}
	return 0;
}
```

```
void factorial(int no)
{
	int fact,i=1;
	if (no==0)
		printf("\nFactorial of the 0 is : 1\n\n");
	else
	{
		fact=no;
		while(i<no)
		{
			fact=fact*i;
			i++;
		}
		printf("\nFactorial of the %d is : %d\n\n",no,fact);
	}
}

void prime(int no)
{
	int clk=sqrt(no);
	int flag;
	for(int i=2;i<=clk;i++)
	{
		if(no%i==0)
		{
			flag=1;break;
		}
		else
			flag=0;
	}
	if(flag==1)
		printf("\n%d is not Prime!\n\n",no);
	else
		printf("\n%d is Prime!\n\n",no);
}

void amstrong(int no)
{
	int disc=no,ams=0;
	while(disc>=10)
	{
		int temp=disc%10;
		ams+=(temp*temp*temp);
		disc/=10;
	}
	ams+=(disc*disc*disc);
	if(ams==no)
		printf("%d is an Amstrong Number!\n\n",no);
	else
		printf("%d is not an Amstrong Number!\n\n",no);
}
```

```
void timestamp()
{
	time_t clk = time(NULL);
	printf("\n%s\n", ctime(&clk));
}
```

>>> **Output :**



- **To perform functions with Array.**

>>> Syntax :

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

void timestamp();
void reverse(int*,int);
void minmax(int*,int);

int main(void)
{
	int Arr[100],n,ch;
	timestamp();
	printf("\nEnter the length of the Array [limit 100] :");
	scanf("%d",&n);
	for(int i=0;i<n;i++)
	{
		printf("\nEnter element at position %d :",i);
		scanf("%d",&Arr[i]);
	}
	printf("\nEntered Array is :\n");
	for(int i=0;i<n;i++)
	{
		printf("%d\t",Arr[i]);
	}
	printf("\n\nTo Perform : \n1. Revesre Order \n2. Min-Max\n");
	printf("Enter your Choice :");
	scanf("%d",&ch);
	switch(ch)
	{
		case 1:
			reverse(Arr,n);
			break;
		case 2:
			minmax(Arr,n);
			break;
	}
	return 0;
}

void reverse(int A[],int n)
{
	printf("\nArray in reverse order is :\n");
	for(int i=n-1;i>=0;i--)
	{
		printf("%d\t",A[i]);
	}printf("\n\n");
}

void minmax(int A[],int n)
{
	int max,min;
```

```
        max=A[0];
        min=A[0];
        for(int i=0;i<n;i++)
        {
                if(max < A[i])
                        max=A[i];

                if(min > A[i])
                        min=A[i];
                else
                        continue;
        }
        printf("Max : %d & MIN : %d\n\n",max,min);
}

void timestamp()
{
        time_t clk = time(NULL);
        printf("\n%s\n", ctime(&clk));
}
```

>>> **Output :**

```
khushal@khushal-Veriton-Series:~/Desktop/assignment$ ./a.out

Sat Sep 15 11:18:48 2018


Enter the length of the Array [limit 100] :12

Enter element at position 0 :14

Enter element at position 1 :12

Enter element at position 2 :15

Enter element at position 3 :1456

Enter element at position 4 :1236

Enter element at position 5 :10

Enter element at position 6 :1234

Enter element at position 7 :142

Enter element at position 8 :1536

Enter element at position 9 :1475

Enter element at position 10 :10236

Enter element at position 11 :10254

Entered Array is :
14      12      15      1456    1236    10      1234    142     1536    1475    10236   10254
To Perform :
1. Revesre Order
2. Min-Max
Enter your Choice :2
Max : 10254 & MIN : 10

khushal@khushal-Veriton-Series:~/Desktop/assignment$
```

```
khushal@khushal-Veriton-Series:~/Desktop/assignment$ ./a.out
Sat Sep 15 11:25:48 2018

Enter the length of the Array [limit 100] :5
Enter element at position 0 :1
Enter element at position 1 :3
Enter element at position 2 :5
Enter element at position 3 :4
Enter element at position 4 :2
Entered Array is :
1       3       5       4       2

To Perform :
1. Revesre Order
2. Min-Max
Enter your Choice :1

Array in reverse order is :
2       4       5       3       1

khushal@khushal-Veriton-Series:~/Desktop/assignment$
```

- **To check whether the input number is a palendrome or not.**

  >>> Syntax :

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

int reverse(int);
void timestamp();
int main(void)
{
        int num,rev;
        timestamp();
        printf("\nEnter the number :");
        scanf("%d",&num);
        rev=reverse(num);
        printf("\nReverse of the Number is : %d\n",rev);
        if(num==rev)
                printf("\n%d is a Palendrome Number!\n\n",num);
        else
                printf("\n%d is not a Palendrome Number!\n\n",num);
        return 0;
}

int reverse(int no)
{
        int segment,palen=0;
        while(no>=10)
        {
                segment=no%10;
```

```
                no/=10;
                palen=(palen*10)+segment;
        }palen=(palen*10)+no;
        return palen;
}

void timestamp()
{
        time_t clk=time(NULL);
        printf("\n%s\n", ctime(&clk));
}
```

>>> **Output :**

```
khushal@khushal-Veriton-Series:~/Desktop/assignment$ ./a.out

Wed Sep 26 18:16:15 2018


Enter the number :1221

Reverse of the Number is : 1221

1221 is a Palendrome Number!

khushal@khushal-Veriton-Series:~/Desktop/assignment$
```

- **To perform various functions with array using Recursion.**

>>> **Syntax :**

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

void timestamp();
int reverse(int[],int);
int display(int[],int,int);
int sum(int[],int);
int main(void)
{
        int Arr[100],len,ch,ret;
        char cont='y';
        timestamp();
        while('Y'==cont||'y'==cont)
        {
                printf("\nEnter the length of the Array [limit 100] :");
                scanf("%d",&len);
                for(int i=0;i<len;i++)
```

```
		{
			printf("\nEnter element at position %d :",i);
			scanf("%d",&Arr[i]);
		}
		printf("\nEntered Array is :\n");
		for(int i=0;i<len;i++)
		{
			printf("%d\t",Arr[i]);
		}
		printf("\n\nMENU ::\n1.Display Array\n2.Array in Reverse Order\n3.Sum of
Elements\nEnter the Choice:");
		scanf("%d",&ch);
		switch(ch)
		{
			case 1:
				printf("\nArray Is:\n");
				display(Arr,len-1,0);
				printf("\n");break;
			case 2:
				printf("\nArray in Reverse order is :\n");
				reverse(Arr,len-1);
				printf("\n");break;
			case 3:
				ret=sum(Arr,len-1);
				printf("\nSum of Elements is :%d\n\n",ret);
		}
		printf("\nDo you want to contunue (Y/N) :");
		scanf("%s",&cont);
	}
	return 0;
}

int display(int Arr[],int n,int str)
{
	if(str>n)
		return 0;
	printf("%d\t",Arr[str]);
	display(Arr,n,str+1);
}

int reverse(int Arr[],int n)
{
	printf("%d\t",Arr[n]);
	if(n==0)
		return 0;
	reverse(Arr,n-1);
}

int sum(int Arr[],int n)
{
	int tot=0;
	if(n<0)
```

```
                return tot;
        tot=Arr[n]+sum(Arr,n-1);
}

void timestamp()
{
        time_t clk=time(NULL);
        printf("\n%s\n",ctime(&clk));
}
```

>>> **Output :**

```
khushal@khushal-Veriton-Series:~/Desktop/assignment$ ./a.out

Wed Sep 26 18:20:39 2018


Enter the length of the Array [limit 100] :5

Enter element at position 0 :1

Enter element at position 1 :3

Enter element at position 2 :5

Enter element at position 3 :4

Enter element at position 4 :2

Entered Array is :
1       3       5       4       2

MENU ::
1.Display Array
2.Array in Reverse Order
3.Sum of Elements
Enter the Choice:2

Array in Reverse order is :
2       4       5       3       1

Do you want to contunue (Y/N) :y

Enter the length of the Array [limit 100] :5

Enter element at position 0 :1

Enter element at position 1 :3

Enter element at position 2 :5
Enter element at position 3 :4

Enter element at position 4 :2

Entered Array is :
1       3       5       4       2

MENU ::
1.Display Array
2.Array in Reverse Order
3.Sum of Elements
Enter the Choice:3

Sum of Elements is :15


Do you want to contunue (Y/N) :n
khushal@khushal-Veriton-Series:~/Desktop/assignment$
```

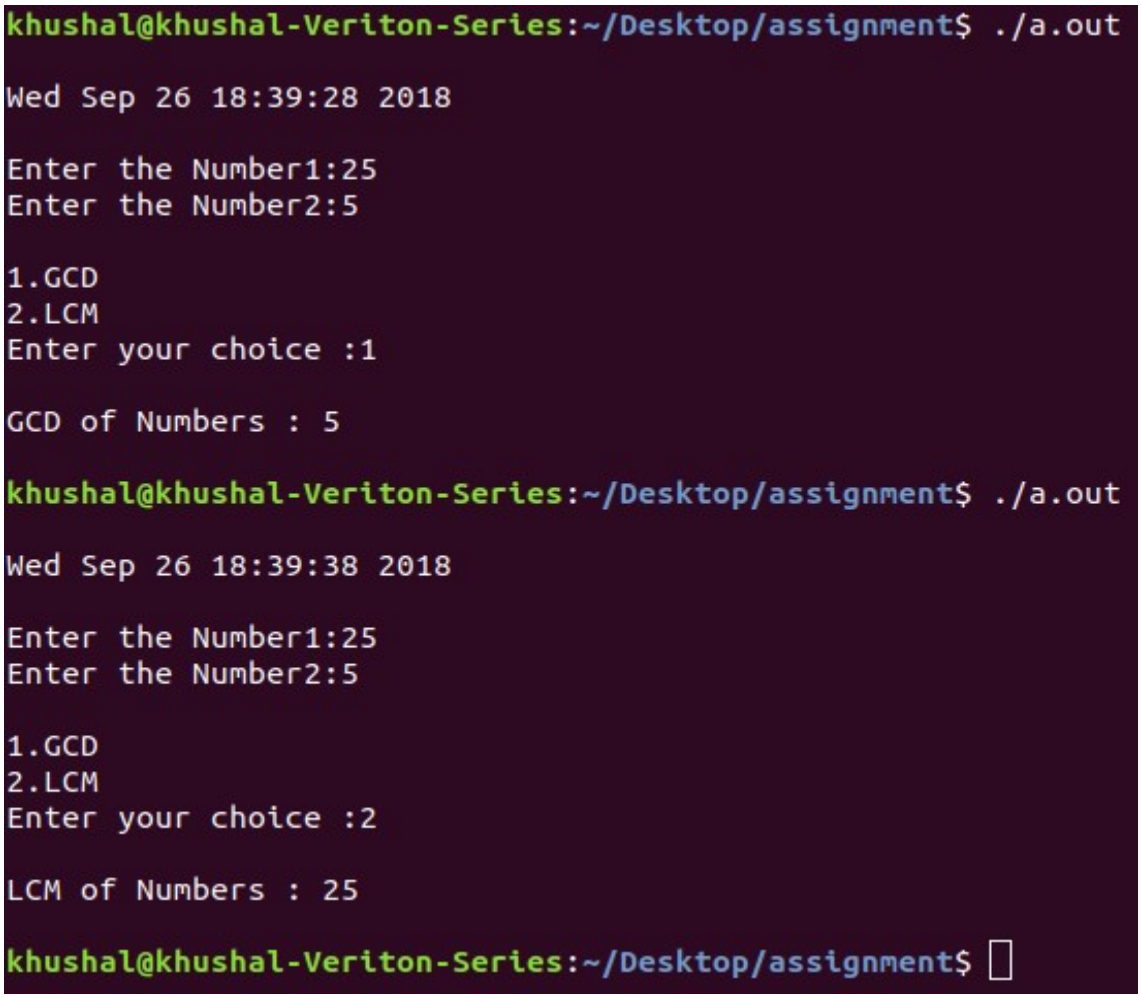
- **To print the GCD & LCM of 2 input numbers using Recursion.**

  >>> Syntax :

```
#include<stdio.h>
#include<time.h>
int i=2,count=1;
void timestamp();
int gcd(int,int);
int lcm(int,int);
int main(void)
{
	int num1,num2,ch;
	timestamp();
	printf("Enter the Number1:");
	scanf("%d",&num1);
	printf("Enter the Number2:");
	scanf("%d",&num2);
	printf("\n1.GCD\n2.LCM\nEnter your choice :");
	scanf("%d",&ch);
	switch(ch)
	{
		case 1:printf("\nGCD of Numbers : %d\n\n",gcd(num1,num2));break;
		case 2:printf("\nLCM of Numbers : %d\n\n",lcm(num1,num2));break;
	}
	return 0;
}

int lcm(int no1,int no2)
{
	if(no1==1 && no2==1)
		return count;
	if(no1%i==0 || no2%i==0)
	{
		count*=i;
		if(no1%i==0)
			no1/=i;
		if(no2%i==0)
			no2/=i;
	}
	else
	{i++;}
	lcm(no1,no2);
}

int gcd(int no1,int no2)
{
	if(no1%i==0 && no2%i==0)
	{
		no1/=i;
		no2/=i;
		count*=i;
		gcd(no1,no2);
	}
```

```
        if(i<=no1 || i<=no2)
        {
                i++;
                gcd(no1,no2);
        }
        else
                return count;
}

void timestamp()
{
time_t clk=time(NULL);
printf("\n%s\n",ctime(&clk));
}
```

>>> **Output :**



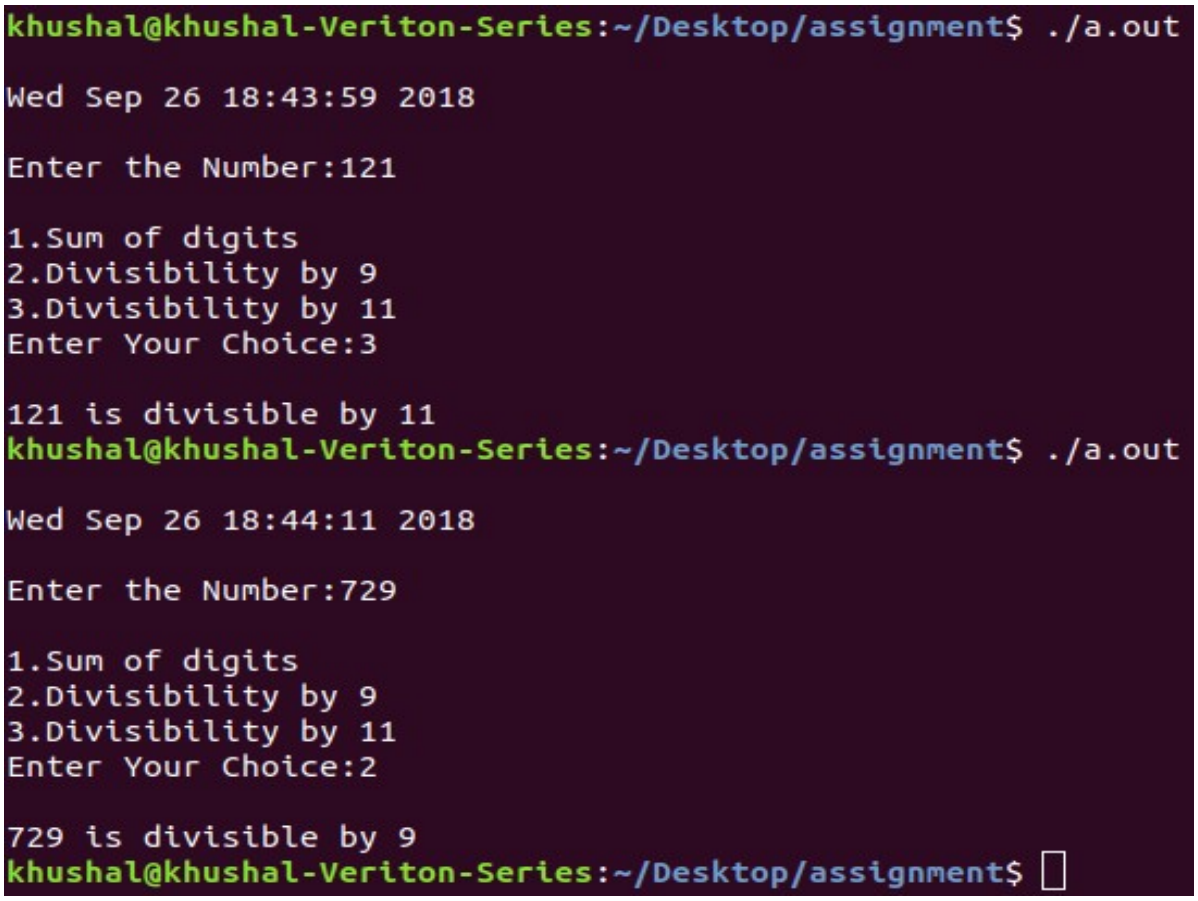
- **To print the sum of digits of the input number and check its' divisibility by 9 and 11 using Recursion.**

>>> Syntax :

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

int a=0,b=0;
void timestamp();
int sum(int);
int nine(int);
int eleven(int);
int main(void)
{
	int num,flag,ch;
	timestamp();
	printf("Enter the Number:");
	scanf("%d",&num);
	printf("\n1.Sum of digits\n2.Divisibility by 9\n3.Divisibility by 11\nEnter Your Choice:");
	scanf("%d",&ch);
	switch(ch){
		case 1:
			printf("\nSum of Digits : %d\n\n",sum(num));break;
		case 2:
			flag=nine(num);
			if(flag==1)
				printf("\n%d is divisible by 9\n",num);
			else
				printf("\n%d is not divisible by 9\n",num);
		break;
		case 3:
			flag=eleven(num);
			if(flag==1)
				printf("\n%d is divisible by 11\n",num);
			else
				printf("\n%d is not divisible by 11\n",num);
		break;
	}
	return 0;
}

int nine(int no)
{
	int count;
	count=sum(no);
	if(count%9==0)
		return 1;
	else
		return 0;
}

int eleven(int no)
{
```

```
        a+=no%10;
        no/=10;
        b+=no%10;
        no/=10;
        if(no<1)
                return;
        eleven(no);
        if(a-b==0 || a-b==11 || b-a==11)
                return 1;
        else
                return 0;
}
int sum(int no)
{
        int count=0;
        if(no<1)
                return;
        count=(no%10)+sum(no/10);
        return count;
}

void timestamp()
{
time_t clk=time(NULL);
printf("\n%s\n",ctime(&clk));
}
```

>>> **Output :**

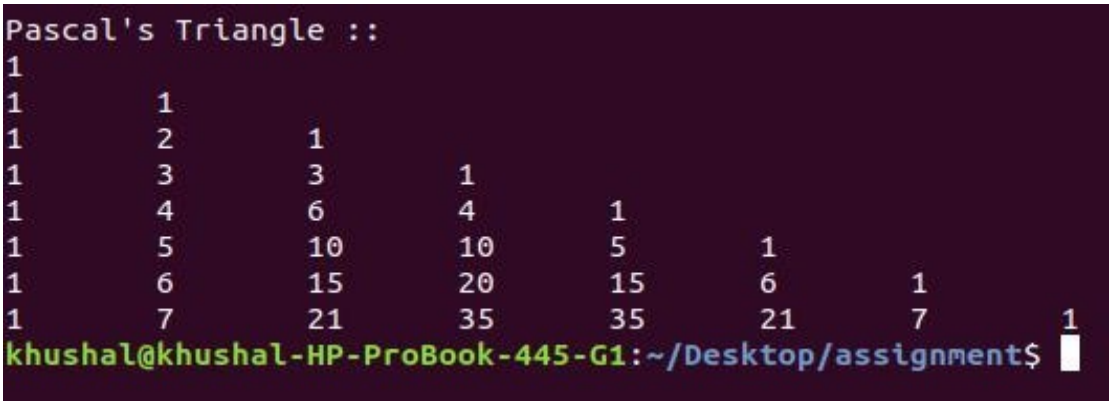


- **To print the Pascal Triangle using a 2-D array.**

>>> Syntax :

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

int len;
int main(void)
{
	printf("Enter the size :");
	scanf("%d",&len);
	int Arr[len][len];
	system("clear");
	for(int i=0;i<len;i++)
		for(int j=0;j<=i;j++)
		{
			if(j == 0 || i == j)
				Arr[i][j]=1;
			else
				Arr[i][j]=Arr[i-1][j]+Arr[i-1][j-1];
		}
	printf("\nPascal's Triangle ::\n");
	for(int i=0;i<len;i++){
		for(int j=0;j<=i;j++)
			printf("%d\t",Arr[i][j]);
		printf("\n");}
	return 0;
}
```

>>> Output :



- **To perform Linear Search and Binary Search using Recursion.**

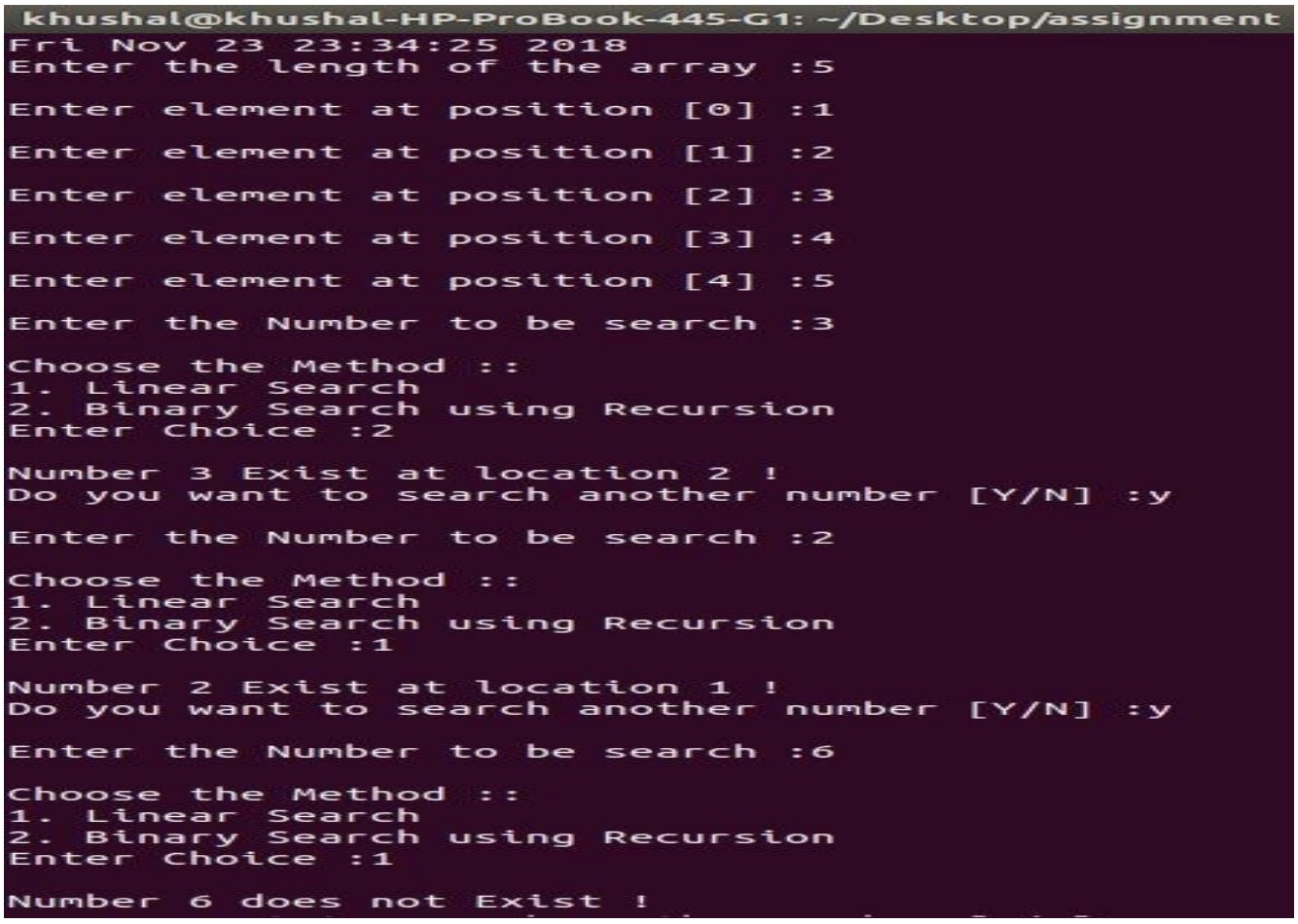
>>> Syntax :

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
int len;
```

```
void timestamp();
int linear(int *,int,int);
int binary(int *,int,int,int,int);
int main(void)
{
	system("clear");
	timestamp();
	int Arr[len],num,ch,flag;
	char ans='Y';
	printf("Enter the length of the array :");
	scanf("%d",&len);
	for(int i=0;i<len;i++)
	{
		printf("\nEnter element at position [%d] :",i);
		scanf("%d",&Arr[i]);
	}
	do
	{
		printf("\nEnter the Number to be search :");
		scanf("%d",&num);
		printf("\nChoose the Method ::\n");
		printf("1. Linear Search\n2. Binary Search using Recursion\n");
		printf("Enter Choice :");
		scanf("%d",&ch);
		switch(ch)
		{
			case 1:
				flag=linear(Arr,num,len);
				if(flag == len+1)
					printf("\nNumber %d does not Exist !",num);
				else
					printf("\nNumber %d Exist at location %d !",num,flag);
				break;
			case 2:
				flag=binary(Arr,num,0,len,len+1);
				if(flag == len+1)
					printf("\nNumber %d does not Exist !",num);
				else
					printf("\nNumber %d Exist at location %d !",num,flag);
				break;
		}
		printf("\nDo you want to search another number [Y/N] :");
		scanf("%s",&ans);
	}while(ans == 'y' || ans == 'Y');
	return 0;
}

int linear(int Arr[],int num,int len)
{
	for(int i=0;i<len;i++)
		if(Arr[i]==num)
			return i;
```

```
        return len+1;
}

int binary(int Arr[],int num,int top,int bot,int len1)
{
        int mid=(top+bot)/2;
        if(Arr[top]==num)
                return top;
        else if(Arr[bot]==num)
                return bot;
        else if(Arr[mid]==num)
                return mid;
        else if(top==bot)
                return len1;
        else if(num>Arr[mid])
                binary(Arr,num,mid,bot,len1);
        else if(num<Arr[mid])
                binary(Arr,num,0,mid,len1);
}

void timestamp()
{
        time_t clk=time(NULL);
        printf("%s",ctime(&clk));
}
```

>>> **Output :**



- **To perform Addition,Subtraction,Multiplication and Transpose of a Matrix.**

>>> Syntax :

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

int len1,len2;

void timestamp();
void input();
void display();
void display1(int,int);
void addition();
void subtraction();
void multiplication();
void transpose();

int A[10][10],B[10][10],C[10][10];
int main()
{

//      int A[][len2],B[len1][len2],C[len1][len2];
        char ans;
start:
        system("clear");
        int ch;
        timestamp();
                printf("\nMENU ::\n");
                printf("1.Addition of matrices\n");
                printf("2.Subtraction of matrices\n");
                printf("3.Multiplication of matrices\n");
                printf("4.Transpose of Matrix\n");
                printf("ENTER YOUR CHOICE:");
                scanf("%d",&ch);
                switch(ch)
                {
                        case 1:
                                printf("\nEnter the number of rows of the matrix : ");
                                scanf("%d",&len1);
                                printf("\nEnter the number of columns of the matrix : ");
                                scanf("%d",&len2);
                                input();
                                display();
                                addition();break;
                        case 2:
                                printf("\nEnter the number of rows of the matrix : ");
                                scanf("%d",&len1);
                                printf("\nEnter the number of columns of the matrix : ");
                                scanf("%d",&len2);
                                input();
                                display();
                                subtraction();break;
```

```
                        case 3:
                                multiplication();break;

                        case 4:
                                printf("\nEnter the number of rows of the matrix : ");
                                scanf("%d",&len1);
                                printf("\nEnter the number of columns of the matrix : ");
                                scanf("%d",&len2);
                                transpose(len1,len2);break;

                }
        printf("\n");
        printf("Do you want to continue [Y/N]:");
        scanf("%s",&ans);
        while(ans == 'y' || ans == 'Y')
                goto start;
}

void input()
{
        printf("Matrix-1 INPUT:\n");
        for(int i=0;i<len1;i++)
                for(int j=0;j<len2;j++)
                {
                        printf("Enter elements at location [%d %d]:",i,j);
                        scanf("%d",&A[i][j]);
                }
        printf("Matrix-2 INPUT:\n");
        for(int i=0;i<len1;i++)
                for(int j=0;j<len2;j++)
                {
                        printf("Enter elements at location [%d %d]:",i,j);
                        scanf("%d",&B[i][j]);
                }
}

void display1(int size1,int size2)
{
        for(int i=0;i<size1;i++)
        {
                for(int j=0;j<size2;j++)
                {
                        printf("%d\t",C[i][j]);
                }
                printf("\n");
        }
        printf("\n");
}
void display()
{
        printf("Entered Matrix-1 is :\n");
        for(int i=0;i<len1;i++)
```

```
	{
		for(int j=0;j<len2;j++)
		{
			printf("%d\t",A[i][j]);
		}
		printf("\n");
	}
	printf("\n");
	printf("Entered Matrix-2 is :\n");
	for(int i=0;i<len1;i++)
	{
		for(int j=0;j<len2;j++)
		{
			printf("%d\t",B[i][j]);
		}
		printf("\n");
	}
	printf("\n");
}

void addition()
{
	for(int i=0;i<len1;i++)
	for(int j=0;j<len2;j++)
	{
		C[i][j]=A[i][j]+B[i][j];
	}
	printf("Sum of matrix are:\n");
	display1(len1,len2);
}

void subtraction()
{
	for(int i=0;i<len1;i++)
	for(int j=0;j<len2;j++)
	{
		C[i][j]=A[i][j]-B[i][j];
	}
	printf("Difference b/w matrices are:\n");
	display1(len1,len2);
}

void multiplication()
{
	int m,n,p,q,i,j;
	printf("Enter the length of 1st matrix [Row x Col]:");
	scanf("%d %d",&m,&n);
	for(int i=0;i<m;i++)
		for(int j=0;j<n;j++)
		{
			printf("Enter elements at location [%d %d]:",i,j);
			scanf("%d",&A[i][j]);
```

```
		}
	printf("Enter the length of 2nd matrix [Row x Col]:");
	scanf("%d %d",&p,&q);
	for(i=0;i<p;i++)
		for(j=0;j<q;j++)
		{
			printf("Enter elements at location [%d %d]:",i,j);
			scanf("%d",&B[i][j]);
		}
	for(i=0;i<m;i++)
	{
		for(j=0;j<n;j++)
		{
			printf("%d\t",A[i][j]);
		}
	printf("\n");
	}
	printf("\n");
	for(i=0;i<p;i++)
	{
		for(j=0;j<q;j++)
		{
			printf("%d\t",B[i][j]);
		}
		printf("\n");
	}
	printf("\n");
	if(n==p)
	{
		for(i=0;i<m;i++)
		{
			for(j=0;j<q;j++)
			{
				C[i][j]=0;
				for(int k=0;k<n;k++)
					C[i][j]=C[i][j]+(A[i][k]*B[k][j]);
			}
		}
		display1(m,q);
	}
	else
		printf("Matrix can't be multiplied!\n");
}

void transpose(int len1,int len2)
{
	printf("Matrix INPUT:\n");
	for(int i=0;i<len1;i++)
		for(int j=0;j<len2;j++)
		{
			printf("Enter elements at location [%d %d]:",i,j);
			scanf("%d",&C[i][j]);
```

```
                }
        printf("Entered Matrix is :\n");
        display1(len1,len2);
        printf("Transpose of matrix is:\n");
        for(int i=0;i<len2;i++)
        {
                for(int j=0;j<len1;j++)
                {
                        printf("%d\t",C[j][i]);
                }
                printf("\n");
        }
}

void timestamp()
{
        time_t clk=time(NULL);
        printf("%s",ctime(&clk));
}
```

>>> **Output :**

```
Fri Nov 23 22:51:29 2018

MENU ::
1.Addition of matrices
2.Subtraction of matrices
3.Multiplication of matrices
4.Transpose of Matrix
ENTER YOUR CHOICE:3
Enter the length of 1st matrix [Row x Col]:2
2
Enter elements at location [0 0]:1
Enter elements at location [0 1]:2
Enter elements at location [1 0]:3
Enter elements at location [1 1]:4
Enter the length of 2nd matrix [Row x Col]:2
3
Enter elements at location [0 0]:1
Enter elements at location [0 1]:2
Enter elements at location [0 2]:3
Enter elements at location [1 0]:4
Enter elements at location [1 1]:5
Enter elements at location [1 2]:6
1       2
3       4

1       2       3
4       5       6

9       12      15
19      26      33


Do you want to continue [Y/N]:y
```

```
Fri Nov 23 22:52:10 2018

MENU ::
1.Addition of matrices
2.Subtraction of matrices
3.Multiplication of matrices
4.Transpose of Matrix
ENTER YOUR CHOICE:4

Enter the number of rows of the matrix : 3

Enter the number of columns of the matrix : 2
Matrix INPUT:
Enter elements at location [0 0]:1
Enter elements at location [0 1]:2
Enter elements at location [1 0]:3
Enter elements at location [1 1]:4
Enter elements at location [2 0]:5
Enter elements at location [2 1]:6
Entered Matrix is :
1       2
3       4
5       6

Transpose of matrix is:
1       3       5
2       4       6

Do you want to continue [Y/N]:n
khushal@khushal-HP-ProBook-445-G1:~/Desktop/assignment$
```

- **To create a 3-D Array and make a function to alter/modify its elements.Also allocate the Array using DMA (Use malloc,memset and free as well.)**

>>> Syntax :

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<time.h>

void timestamp();
void display(int (*)[10][10],int,int,int);
int main(void)
{
        int (*A)[10][10],len1,len2,len3;
        A = (int (*)[10][10]) malloc(len1*len2*len3*sizeof(int));
        memset(A,0,len1*len2*len3*sizeof(int));
        char ans,ans1='Y';
        timestamp();
```

```
	printf("\nEnter the number of rows of the 3-D matrix : ");
	scanf("%d",&len1);
	printf("\nEnter the number of rows of the 2-D matrix : ");
	scanf("%d",&len2);
	printf("\nEnter the number of columns of the 2-D matrix : ");
	scanf("%d",&len3);
	for(int i=0;i<len1;i++){
		printf("\n%d 2-D Matrix input ::\n",i);
		for(int j=0;j<len2;j++)
		for(int k=0;k<len3;k++)
		{
			printf("Enter elements at location [%d %d %d]:",i,j,k);
			scanf("%d",(*(*(A+i)+j)+k));
		}printf("\n");
	}
	display(A,len1,len2,len3);
	getchar();
	printf("\nDo you want to alter Matrix [Y/N]:");
	scanf("%c",&ans);
	while(ans == 'Y' || ans == 'y'){
	while(ans1 == 'Y' || ans1 == 'y')
	{
		int i=0,j=0,k=0;
		printf("Enter the number of 2-D Matrix :");
		scanf("%d",&i);
		printf("Enter the row of the element :");
		scanf("%d",&j);
		printf("Enter the column of the element :");
		scanf("%d",&k);
		if(i>=len1 || j>=len2 || k>=len3)
			printf("\nInvalid Position!\n");
		else
		{
			printf("Enter Number :");
			scanf("%d",(*(*(A+i)+j)+k));
		}
		getchar();
		printf("\nDo you want to modify another [Y/N]:");
		scanf("%c",&ans1);
	}ans=ans1;
	display(A,len1,len2,len3);
	}
	free(A);
	return 0;
}

void display(int (*A)[10][10],int len1,int len2,int len3)
{
	for(int i=0;i<len1;i++)
	{
		printf("%d 2-D Matrix output ::\n",i);
		for(int j=0;j<len2;j++)
```

```
                {
                        for(int k=0;k<len3;k++)
                        {
                                printf("%d\t",*(*(*(A+i)+j)+k));
                        }
                        printf("\n");
                }
                printf("\n\n\n");
        }
}

void timestamp()
{
        time_t clk=time(NULL);
        printf("%s",ctime(&clk));
}
```

>>> **Output :**

```
khushal@khushal-HP-ProBook-445-G1:~/Desktop/assignment$ gcc three.c
khushal@khushal-HP-ProBook-445-G1:~/Desktop/assignment$ ./a.out
Fri Nov 23 23:06:29 2018

Enter the number of rows of the 3-D matrix : 2

Enter the number of rows of the 2-D matrix : 2

Enter the number of columns of the 2-D matrix : 2

0 2-D Matrix input ::
Enter elements at location [0 0 0]:1
Enter elements at location [0 0 1]:2
Enter elements at location [0 1 0]:3
Enter elements at location [0 1 1]:4


1 2-D Matrix input ::
Enter elements at location [1 0 0]:5
Enter elements at location [1 0 1]:6
Enter elements at location [1 1 0]:7
Enter elements at location [1 1 1]:8

0 2-D Matrix output ::
1       2
3       4



1 2-D Matrix output ::
5       6
7       8
```

```
Do you want to alter Matrix [Y/N]:y
Enter the number of 2-D Matrix :1
Enter the row of the element :1
Enter the column of the element :2

Invalid Position!

Do you want to modify another [Y/N]:y
Enter the number of 2-D Matrix :1
Enter the row of the element :0
Enter the column of the element :0
Enter Number :10

Do you want to modify another [Y/N]:n
0 2-D Matrix output ::
1       2
3       4



1 2-D Matrix output ::
10      6
7       8



khushal@khushal-HP-ProBook-445-G1:~/Desktop/assignment$
```

- **To create a customized String function (Eg. Strlen, strcpy, strcat, strcmp).**

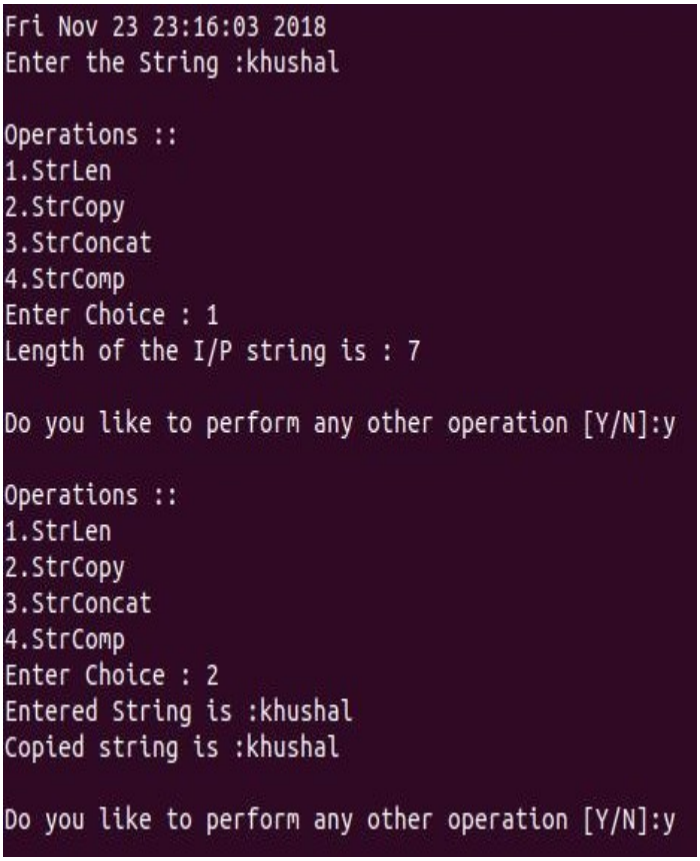
>>> Syntax :

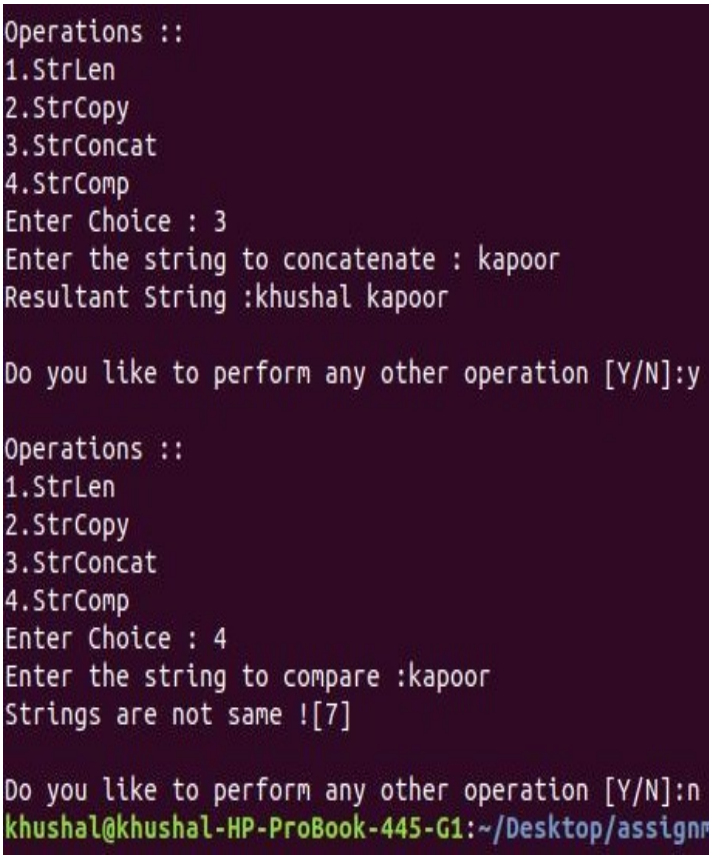
```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<time.h>

int strlength(char *);
void timestamp();
char *strcopy(char*,char*);
char *strconcat(char*,char*);
int strcomp(char*,char*);
int main(void)
{
        system("clear");
        timestamp();
        char string[100],string1[100];
        int ch,flag;
```

```
	char *p,ans;
	printf("Enter the String :");
	fgets(string,sizeof(string),stdin);
	do{
		printf("\nOperations ::\n");
		printf("1.StrLen \n2.StrCpy \n3.StrCat \n4.StrComp\n");
		printf("Enter Choice : ");
		scanf("%d",&ch);
		getchar();
		switch(ch)
		{
			case 1:
				printf("Length of the I/P string is : %d\n",strlength(string)-1);break;
			case 2:
				printf("Entered String is :");
				printf("%s",string);
				p=strcopy(string,string1);
				printf ("Copied string is :");
				printf("%s",p);
				break;
			case 3:
				printf("Enter the string to concatenate :");
				//scanf("%s",string1);
				fgets(string1,sizeof(string1),stdin);
				printf("Resultant String :");
				p=strconcat(string,string1);
				printf("%s",p);
				break;
			case 4:
				printf("Enter the string to compare :");
				fgets(string1,sizeof(string1),stdin);
				flag=strcomp(string,string1);
				if(flag == 0)
					printf("Strings are same !\n");
				else
					printf("Strings are not same ![%d]\n",flag);
				break;
		}
		printf("\nDo you like to perform any other operation [Y/N]:");
		scanf("%c",&ans);
	}while(ans == 'Y' || ans == 'y');
	return 0;
}

int strlength(char string[])
{
	int i=0;
	while(string[i]!='\0')
		i++;
	return i;
}
```

```
char *strcopy(char string[],char string1[])
{
	int i=0;
	while(string[i]!='\0')
	{
		string1[i]=string[i];
		i++;
	}
	return string1;
}

char *strconcat(char string[],char string1[])
{
	int i=strlength(string)-1;
	int j=0;
	while(string1[j]!='\0')
	{
		string[i]=string1[j];
		i++;
		j++;
	}
	string[i]='\0';
	return string;
}

int strcomp(char string[],char string1[])
{
	int i=0;
	while(string[i] == string1[i])
	{
		i++;
		if(string[i] == '\0')
			return 0;
		continue;
	}
	return string[i]-string1[i];
}

void timestamp()
{
	time_t clk=time(NULL);
	printf("%s",ctime(&clk));
}
```

>>> **Output :**





- **To create a String function to reverse a string word-by-word.**

>>> **Syntax :**

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<time.h>

char str[100];
void timestamp();
char *strrev(char*,int);
int main(void)
{
        system("clear");
        timestamp();
        char string[100],*p;
        printf("\nEnter the string :");
        fgets(string,sizeof(string),stdin);
        printf("\nReverse of the string is :: \n");
        p=strrev(string,strlen(string));
        printf("%s\n",p);
```

```
	return 0;
}

char *strrev(char *string,int size)
{
	int j=-1,k=0;
	string[-1]=' ';
	for(int i=size-1;i>=-1;i--)
	{
		int l=1;
		if(*(string+i) != ' ')
			j++;
		if(*(string+i) == ' ')
		{
			while(j>0)
			{
				*(str+k)=*(string+i+l);
				j--;k++;l++;
			}
			*(str+k)=' ';
			k++;
		}
	}
	return str;
}

void timestamp()
{
	time_t clk=time(NULL);
	printf("%s",ctime(&clk));
}
```

>>> **Output :**

```
Fri Nov 23 23:26:08 2018

Enter the string :My name is Khushal Kapoor

Reverse of the string is ::
Kapoor Khushal is name My
khushal@khushal-HP-ProBook-445-G1:~/Desktop/assignment$
```