



# APPENDIX

## ADDITIONAL C PROGRAMS

### 1. Find the sum of two numbers.

```
#include<stdio.h>
int main()
{
    int a, b, c;
    printf("Enter the values of a and b \n");
    scanf("%d%d", &a, &b);
    c = a + b;
    printf("Sum of a and b is = %d\n", c);
    return 0;
}
```

#### Output

```
Enter the values of a and b
5
6
Sum of a and b is = 11
```

### 2. Print the 'Hello World' in C.

```
//C hello world example
#include <stdio.h>

int main()
{
    printf("Hello world\n");
    return 0;
}
```

**Output**

Hello world

**3. Write a program to display an integer.**

```
#include <stdio.h>
int main()
{
    int a;
    printf("Enter the value of a \n");
    scanf("%d", &a);
    printf("The Entered value is %d\n", a);
    return 0;
}
```

**Output**

Enter the value of a  
5  
The entered value is 5

**4. Find whether the given number is odd or even.**

```
#include <stdio.h>
int main()
{
    int n;

    printf("Enter the value of n \n");
    scanf("%d", &n);

    if (n%2 == 0)
        printf("The number is Even\n");
    else
        printf("The number is Odd\n");
    return 0;
}
```

**Output**

Enter the value of n  
6  
The number is Even

**5. Program to perform arithmetic operations (addition, subtraction, multiplication, division).**

```
#include <stdio.h>
int main()
{
    int a, b, add, subtract, multiply;
    float divide;
```

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

add = a+b;
subtract = a-b;
multiply = a*b;
divide = a/ (float)b;    //typecasting

printf("Sum = %d\n",add);
printf("Difference = %d\n",subtract);
printf("Multiplication = %d\n",multiply);
printf("Division = %.2f\n",divide); // %.2f is used to display the two decimal numbers.

return 0;
}
```

**Output**

```
Enter the values of a and b
4
7
Sum = 11
Difference = -3
Multiplication = 28
Division = 0.57
```

**6. Check whether the given year is leap year or not.**

```
#include <stdio.h>

int main()
{
    int year;

    printf("Enter a year \n");
    scanf("%d", &year);

    if ( year % 400 == 0)
        printf("%d is a leap year.\n", year);
    else if ( year % 100 == 0)
        printf("%d is not a leap year.\n", year);
    else if ( year % 4 == 0 )
        printf("%d is a leap year.\n", year);
    else
        printf("%d is not a leap year.\n", year);

    return 0;
}
```

**Output**

```
Enter a year
1996
1996 is a leap year.
```

**7. Find the factorial.**

```
#include <stdio.h>

int main()
{
    int c, n, fact = 1;
    printf("Enter a number \n");
    scanf("%d", &n);

    for (c = 1; c <= n; c++)
        fact = fact * c;

    printf("Factorial of given number %d is = %d\n", n, fact);
    return 0;
}
```

**Output**

```
Enter a number
8
Factorial of the given number 8 is = 40320
```

**8. Check whether the given number is perfect number or not.**

```
#include<stdio.h>
int main(){
    int n,i=1,sum=0;
    printf("Enter a number: ");
    scanf("%d",&n);

    for(i=0;i<n;i++){
        if(n%i==0)
            sum=sum+i;
    }
    if(sum==n)
        printf(" The Entered number %d is a perfect number ",n);
    else
        printf("The Entered number %d is not a perfect number",n);
    return 0;
}
```

**Output**

```
Enter a number: 6
The entered number 6 is a perfect number
```

**9. Write a program to illustrate the simple calculator using switch statement.**

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

```
void main()
{
    int a, b, result, n;
    clrscr();
    printf("\n\t\t\t\t MENU");
    printf("\n\n\t\t\t1.Addition");
    printf("\n\n\t\t\t2.Subtraction");
    printf("\n\n\t\t\t3.Multiplication");
    printf("\n\n\t\t\t4.Division");
    printf("\n\n\t\t\t5.Exit");
    printf("\n\n\t\t6.Enter the values of a and b : ");
    scanf("%d%d",&a,&b);
    printf("\n\n\t\t Enter your choice:");
    scanf("%d",&n);

    switch(n)
    {
        case 1:
            result=a+b;
            printf("Addition of two numbers=%d",result);
            break;
        case 2:
            result=a-b;
            printf("Subtraction of two numbers=%d",result);
            break;
        case 3:
            result=a*b;
            printf("Multiplication of two numbers=%d",result);
            break;
        case 4:
            result=a/b;
            printf("Division of two numbers=%d",result);
            break;
        case 5:
            exit(0);
    }
    getch();
}
```

**Output**

MENU

1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit

```

Enter two numbers
4
2

Enter your choice: 1
Addition of two numbers= 6
Enter your choice: 2
Subtraction of two numbers= 2

Enter your choice: 3
Multiplication of two numbers= 8

Enter your choice: 4
Division of two numbers= 2

Enter your choice: 5

```

### 10. Display the Pascal's triangle.

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int binom,p,q,r,x;
    binom=1;
    q=0;
    clrscr();
    printf("\nEnter the number of rows for pascal triangle:");
    scanf("%d",&p);
    printf("\n The Pascal's Triangle is \n");
    while(q<p)
    {
        for(r=40-3*q;r>0;--r)
            printf(" ");
        for(x=0;x<=q;++x)
        {
            if((x==0) || (q==0))
                binom=1;
            else
                binom=(binom*(q-x+1))/x;
            printf("%6d",binom);
        }
        printf("\n");
        ++q;
    }
    getch();
}

```

#### Output

```

Enter the number of rows for Pascal's triangle: 3

```

The Pascal's triangle is

```
1
1   1
1   2   1
```

### 11. Find the multiplication table for a given number.

```
#include<stdio.h>
#include<stdio.h>
void main()
{
    int i,tno,limit;
    limit=1;
    clrscr();
    printf("\nEnter the table number needed and the range :");
    scanf("%d %d",&tno,&limit);
    printf("\nThe Multiplication table for %d is \n",tno);
    for(i=1;i<=limit;i++)
    {
        printf("\n %dx%d=%d\n",tno,i,tno*i);
    }
    getch();
}
```

#### Output

```
Enter the table number needed and the range
3
4
```

```
The multiplication table for 3 is
3x1 = 3
3x2 = 6
3x3 = 9
3x4 = 12
```

### 12. Find the sum of 'n' numbers using function.

```
main()
{
    int sum(int);
    int n;
    clrscr();
    printf("Enter value of n :");
    scanf("%d",&n);
    printf("\n The Sum of %d number is = %d",n,sum(n));
}
int sum(int n)
```

```
{
int i,a,s=0;
for(i=0;i<n;i++)
{
scanf("%d",&a);
s=s+a;
}
return s;
}
```

**Output**

```
Enter value of n
3
1
2
3
The um of 3 numbers is = 6
```

**13. Find the second largest digit from a number using function.**

```
int largest(long int );
void main()
{
long int num;
clrscr();
printf("Enter the number:");
scanf("%ld",&num);
printf("The Second Largest digit in a Number sequence is: %d",largest(num));
getch();
}

int largest(long int num)
{
int n,i,j,temp,a[10];
n=0;

while(num>0)
{
a[n]=num%10;
num=num/10;
n++;
}

//Array Sorting(descending order)
for(i=0;i<n;i++)
{
for(j=i+1;j<n;j++)
{
if(a[i]<a[j])
```



```

{
temp=a[i];
a[i]=a[j];
a[j]=temp;
}
}
}
return(a[1]);
}

```

**Output**

Enter the number

1543

The second largest digit in a number sequence is 4

**14. Find the sum of digits of a number using function.**

```

int sum(long int );
void main()
{
long int n;
clrscr();
printf("Enter the value of n:");
scanf("%ld",&n);
printf("Sum of the digits of a given number is : %d",sum(n));
getch();
}

int sum(long int n)
{
int s=0;
while(n>0)
{
s=s+n%10;
n=n/10;
}
return(s);
}

```

**Output**

Enter the value of n

73

Sum of the digits of a given number is: 10

**15. Write a C program to check whether the given number and its reverse are same.**

```

void sum(long int );
void main()
{

```

```
long int n;
clrscr();
printf("Enter the number:");
scanf("%ld",&n);
sum(n);
getch();
}

void sum(long int n)
{
int num,s=0;
num=n;
while(n>0)
{
s=s*10+n%10;
n=n/10;
}
if(num==s)
printf("The number is a palindrome");
else
printf("The number is not a palindrome");
}
```

**Output**

Enter the number: 537  
The number and its reverse are not same.

Enter the number: 121  
The number and its reverse are same.

**16. Find the sum of square of individual digits of a number using function.**

```
main()
{
int square(int);
int n;
clrscr();
printf("Enter the Number:");
scanf("%d",&n);
printf("\nSum of square of individual digits is: %d",square(n));
getch();
}

int square(int n)
{
int d,s=0;
while(n>0)
{
```

```
d=n%10;
s=s+d*d;
n=n/10;
}
return s;
}
```

**Output**

```
Enter the number
2
```

Sum of square of individual digits is 4

**17. Write a C program using pointers to swap two numbers.**

```
#include<stdlib.h>
int main()
{
    int *p1=(int*)malloc(sizeof(int)),*p2=(int*)malloc(sizeof(int)),*temp=(int*)
    malloc(sizeof(int));
    printf("Enter two number:");
    scanf("%d%d",p1,p2);
    *temp=*p1;
    *p1=*p2;
    *p2=*temp;
    printf("\nAfter swapping the values:");
    printf("\np1=%d\np2=%d",*p1,*p2);
    return 0;
}
```

**Output**

```
Enter two numbers: 10 15
After swapping the values
p1 = 15
p2 = 10
```

**18. Find the minimum and maximum of a set of numbers.**

```
main()
{
    void num(int);
    int n;
    clrscr();
    printf("Enter n value:");
    scanf("%d",&n);
    num(n);
}
void num(int n)
{
```

```
int i,a,min,max;
for(i=0;i<n;i++)
{
scanf("%d",&a);
if(min>a)
min=a;
if(max<a)
max=a;
}
printf("\nMinmum Value is =%d",min);
printf("\nMaximum Value is=%d",max);
}
```

**Output**

```
Enter n value
5
3
2
4
5
6

Min value = 2
Max value = 6
```

**19. Sort 'n' numbers in ascending order using dynamic memory allocation.**

```
#include<malloc.h>
void main()
{
int n,*a;
int i,j,temp;
clrscr();
printf("Enter the value of n :");
scanf("%d",&n);
a=(int *)malloc(n*sizeof(int));
for(i=0;i<n;i++)
scanf("%d",&a[i]);
for(i=0;i<n;i++)
{
for(j=i+1;j<n;j++)
{
if(a[i]>a[j]) //( For descending order, change the operator as '<' )
{
temp=a[i];
a[i]=a[j];
a[j]=temp;
}
```

```

    }
    }
    }
    for(i=0;i<n;i++)
    printf("%d\t",a[i]);
    getch();
}

```

### Output

```

Enter the value of n
3
2
6
4
2      4      6

```

## 20. Sort 'n' names in ascending order using dynamic memory allocation.

```

#include<stdio.h>
#include<string.h>
#include<stdlib.h>
int main()
{
    char *s[25],temp[25];
    int n,i,j,len;
    printf("enter the number of names to be entered");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        scanf("%s",temp);
        *(s+i)=(char*)malloc(strlen(temp));
        strcpy(*(s+i),temp);
    }

    for(i=0;i<n;i++)
    {
        for(j=i+1;j<n;j++)
        {
            if(strcmp(*(s+i),*(s+j))> 0)    //( For descending order, change the operator as '<' )
            {
                strcpy(temp,*(s+i));
                strcpy(*(s+i),*(s+j));
                strcpy(*(s+j),temp);
            }
        }
    }
}

```

```

}
printf("\n the sorted names  are \n");
for(i=0;i<n;i++)
printf("\n%s",*(s+i));
return 0;
}

```

**Output**

```

Enter the number of names to be entered: 4
Antony
Bala
Arun
Kishore

The sorted names are
Antony
Arun
Bala
Kishore

```

**21. Find the sum of diagonal elements in a matrix.**

```

#include<stdio.h>
int main(){
int a[10][10],i,j,sum=0,m,n;

printf("\nEnter the row and column of matrix: ");
scanf("%d %d",&m,&n);

printf("\nEnter the value  of matrix: ");
for(i=0;i<m;i++)
for(j=0;j<n;j++)
scanf("%d",&a[i][j]);

for(i=0;i<m;i++)
for(j=0;j<n;j++)
if(i==j)
sum=sum+a[i][j];

printf("\n\nSum of the diagonal elements of a entered matrix is: %d",sum);

return 0;
}

```

**Output**

```

Enter the row and column of matrix: 3 3
Enter the value of matrix
2      3      5
6      7      9
2      6      7
Sum of the diagonal elements of the entered matrix is 16.

```

**22. Write a C program to add two matrices using dynamic memory allocation.**

```

#include<malloc.h>
void main()
{
    int m,n,r,s,*a,*b,*c;
    int i,j;
    //clrscr();
    printf("\nEnter the size of FIRST Matrix:");
    scanf("%d%d",&m,&n);
    printf("\nEnter the size of SECOND Matrix:");
    scanf("%d%d",&r,&s);
    if(m!=r || n!=s)
        printf("Matrix Addition is not possible!!!.");
    else
    {
        a=(int*)calloc(m*n ,sizeof(int));
        b=(int*)calloc(r*s,sizeof(int));
        c=(int*)calloc(m*s,sizeof(int));
        //Read the values for a Matrix
        for(i=0;i<m;i++)
        {
            for(j=0;j<n;j++)
                scanf("%d",&a[i*m+j]);
        }
        //Read the values for b Matrix
        for(i=0;i<r;i++)
        {
            for(j=0;j<s;j++)
                scanf("%d",&b[i*r+j]);
        }
        //Adding the matrix a & b
        for(i=0;i<r;i++)
        {
            printf("\n");
            printf("\n The addition of two matrix is:");
            for(j=0;j<s;j++)
            {
                c[i*r+j]=a[i*r+j]+b[i*r+j];
                printf("%d\t",c[i*r+j]);
            }
        }
    }
}

```

**Output**

```
Enter the size of FIRST Matrix: 3 3
Enter the size of SECOND Matrix: 3 3

1 2 3
4 5 6
7 8 9

1 2 3
4 5 6
7 8 9

The addition of two matrices
2 4 6
8 10 12
14 16 18
```

**23. Find the transpose of a given matrix using dynamic memory allocation.**

```
#include<malloc.h>
void main()
{
    int m,n,*a,*b,i,j;
    clrscr();
    printf("\nEnter the size of the Matrix:");
    scanf("%d%d",&m,&n);
    a=(int*)calloc(m*n,sizeof(int));
    b=(int*)calloc(n*m,sizeof(int));

    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            scanf("%d",&a[i*m+j]);
        }
    }
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            b[j*n+i]=a[i*m+j];
        }
    }
    for(i=0;i<n;i++)
    {
        printf("\n");
        for(j=0;j<m;j++)
        {
```



```
printf("%d\t",b[i*n+j]);
}
}
}
```

**Output**

```
Enter the size of the Matrix: 3 2
1 2
3 4
5 6

1 3 5
2 4 6
```

**24. Write a C program to copy one matrix to another matrix using dynamic memory allocation.**

```
#include<malloc.h>
void main()
{
    int m,n,*a,*b;
    int i,j;
    clrscr();
    printf("\nEnter the size of the Matrix:");
    scanf("%d%d",&m,&n);
    a=(int*)calloc(m*n,sizeof(int));
    b=(int*)calloc(m*n,sizeof(int));

    //Read the values for a Matrix
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
            scanf("%d",&a[i*m+j]);
    }

    //Copy the values of 'A' matrix to 'B' Matrix
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
            b[i*m+j]=a[i*m+j];
    }

    //Values of 'B' matrix
    for(i=0;i<m;i++)
    {
        printf("\n");
        for(j=0;j<n;j++)
            printf("%d\t",b[i*m+j]);
    }
}
```

```
getch();  
}
```

**Output**

```
Enter the size of the matrix: 2 2  
1 2  
3 4  
  
1 2  
3 4
```

**25. Find the product of two matrices using dynamic memory allocation.**

```
#include<malloc.h>  
void main()  
{  
    int m,n,*a,*b;  
    int i,j;  
    printf("\nEnter the size of the Matrix:");  
    scanf("%d%d",&m,&n);  
    a=(int*)calloc(m,n*sizeof(int));  
    b=(int*)calloc(m,n*sizeof(int));  
  
    //Read the values for a Matrix  
    for(i=0;i<m;i++)  
    {  
        for(j=0;j<n;j++)  
            scanf("%d",&a[i*m+j]);  
    }  
    //Copy the values of 'A' matrix to 'B' Matrix  
    for(i=0;i<m;i++)  
    {  
        for(j=0;j<n;j++)  
            b[i*m+j]=a[i*m+j];  
    }  
    //Values of 'B' matrix  
    for(i=0;i<m;i++)  
    {  
        printf("\n");  
        for(j=0;j<n;j++)  
            printf("%d\t",b[i*m+j]);  
    }  
}
```

**Output**

```
Enter the size of the Matrix: 2 2  
1 2  
3 4
```

```

1 2
3 4
7 10
15 22

```

**26. Find the sum of elements of a matrix using dynamic memory allocation.**

```

#include<malloc.h>
void main()
{
    int m,n,*a;
    int i,j,sum=0;
    printf("\nEnter the size of the Matrix:");
    scanf("%d%d",&m,&n);
    a=(int*)calloc(m,n*sizeof(int));
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            scanf("%d",&a[i*m+j]);
            sum=sum+a[i*m+j];
        }
    }
    printf("Sum of two matrix is:%d",sum);
}

```

**Output**

```

Enter the size of the matrix 2 2
1 2
3 4
Sum of two matrices is10

```

**27. Write a C program to develop salary details of 5 employees using dynamic memory allocation.**

```

struct employee
{
    int empno;
    char name[20];
    int basic;
    float hra,da,pf,net;
};
struct employee emp,*e;
main()
{
    int i,n;
    clrscr();
    printf("Enter the number of employees:");
}

```

```

scanf("%d",&n);
e=calloc(n,sizeof(emp));
for(i=0;i<n;i++)
{
printf("\n Enter the empno ,name,basic");
scanf("%d%s%d",&e[i].empno,e[i].name,&e[i].basic);
e[i].hra=e[i].basic*.15;
e[i].da=e[i].basic*.35;
e[i].pf=e[i].basic*.10;
e[i].net=e[i].basic+e[i].hra+e[i].da-e[i].pf;
}
printf("\nEmpno\tEmpName\tBasic\tHra\tDa\tPF\tnetSalary");
for (i=0;i<n;i++)
{
printf("\n%d\t%s\t%d\t%.2f\t%.2f\t%.2f\t%.2f",e[i].empno,e[i].name,e[i].basic,e[i].
hra,e[i].da,e[i].pf,e[i].net);
}
getch();}

```

#### Output

Enter the number of employees: 2

Enter the Emp no, name, basic

3

4

5

6

7

8

Empno	EmpName	Basic	Hra	Da	PF	netSalary
3	4	5	0.75	1.75	0.50	7.00
6	7	8	1.20	2.80	0.80	11.20

#### 28. Write a C program to develop a mark sheet of 5 students using dynamic memory allocation.

```

struct student
{
int rollno;
char name[20];
int m1,m2,m3,total;
};
struct student stu,*s;
main()
{
int i,n;
clrscr();
printf("Enter the number of Students:");

```

```

scanf("%d",&n);
s=calloc(n*sizeof(stu));
for(i=0;i<n;i++)
{
scanf("%d%s%d%d",&s[i].rollno,s[i].name,&s[i].m1,&s[i].m2,&s[i].m3);
s[i].total=s[i].m1+s[i].m2+s[i].m3;
}
printf("\nRollno\tName\tMark1\tMark2\tMark3\tTotal");
for (i=0;i<n;i++)
{
printf("\n%d\t%s\t%d\t%d\t%d\t%d",s[i].rollno,s[i].name,s[i].m1,s[i].m2,s[i].m3,s[i].total);
}
getch();
}

```

**Output**

Enter the number of students: 2

```

100    Abishek  95    85    75
101    Logesh   75    85    90

```

Rollno	Name	Mark1	Mark2	Mark3	Total
100	Abishek	95	85	75	255
101	Logesh	75	85	90	250

**29. Write a C program to create a file which contains register number, name, age etc.**

```

#include<stdio.h>
main()
{
FILE *fp;
int i,n;
int rollno;
char name[20];
char dept[10];
int age;
clrscr();
fp=fopen("stu","w");
printf("Enter number of Students:");
scanf("%d",&n);
printf("Rollno\tName\tDept\tAge\n");
for(i=0;i<n;i++)
{
scanf("%d%s%s",&rollno,name,dept,&age);
fprintf(fp,"%d\t%s\t%s\t%d\n",rollno,name,dept,age);
}
fclose(fp);
}

```

```

fp=fopen("stu","r");
printf("\nThe File contents are:");
printf("\n\nRollno\tName\tDept\tAge\n");
for(i=0;i<n;i++)
{
fscanf(fp,"%d%s%s%d",&rollno,name,dept,&age);
printf("\n%d\t%s\t%s\t%d",rollno,name,dept,age);
}
getch();
}

```

### Output

```

Enter the number of students: 2
Rollno      Name      Dept      Age
101         Arun       CSE       19
102         Akshaya   CSE       20

The file contents are

Rollno      Name      Dept      Age
101         Arun       CSE       19
102         Akshaya   CSE       20

```

### 30. Write a C program to copy a file.

```

#include<stdio.h>
main()
{
FILE *fp,*fp1,*fp2;
int i,n;
int rollno;
char name[20];
clrscr();
//Creating a file
fp=fopen("file1","w");
printf("Enter number of Students:");
scanf("%d",&n);
printf("Rollno\tName:\n");
for(i=0;i<n;i++)
{
scanf("%d%s",&rollno,name);
fprintf(fp,"%d\t%s\n",rollno,name);
}
fclose(fp);
//Copy the contents of file from file1 to file2
fp1=fopen("file1","r");
fp2=fopen("file2","w");
for(i=0;i<n;i++)

```

```

{
fscanf(fp1,"%d%s",&rollno,name);
fprintf(fp2,"%d\t%s\n",rollno,name);
}
fclose(fp1);
fclose(fp2);
printf("\n\n The File1 contents are copied to File2...");
//Display the contents of file2
fp2=fopen("file2","r");
rewind(fp2);
printf("\n\n The File2 contents are:");
printf("\nRollno\tName");
for(i=0;i<n;i++)
{
fscanf(fp2,"%d%s",&rollno,name);
printf("\n%d\t%s",rollno,name);
}
getch();}

```

**Output**

```

Enter the number of students: 2
Rollno   Name
100      Antony
101      Bala

The File1 contents are copied to File2

The File2 contents are
Rollno   Name
100      Antony
101      Bala

```

**31. Write a C program to reallocate the memory for the string.**

```

main()
{
char *p;
clrscr();
p=(char *)malloc(7);
printf("\n Before reallocating the memory");
strcpy(p,"kishore");
printf("\n%s",p);
p=(char*)realloc(p,12);
printf("\nAfter reallocating the memory");
strcpy(p,"kishorekumar");
printf("\n%s",p);
getch();
}

```

**Output**

Before reallocating the memory  
kishore  
After reallocating the memory  
kishorekumar

**32. Find whether the given number is a palindrome or not.**

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n,t,s,r;
    clrscr();
    printf("\nEnter the value of n:");
    scanf("%d",&n);
    t=n;
    s=0;
    do
    {
        r=n%10;
        s=(s*10)+r;
        n=n/10;
    }
    while (n>0);
    if (t==s)
        printf ("The given number is a palindrome");
    else
        printf ("The given number is not a palindrome");
    getch();
}
```

**Output**

Enter the number: 121  
The given number is a palindrome.

**33. Write a C program to print a diamond.**

```
#include <stdio.h>
int main(){
    int n, c, k, space = 1;

    printf("Enter number of rows\n");
    scanf("%d", &n);

    space = n - 1;
    for (k = 1; k <= n; k++)
    {
        for (c = 1; c <= space; c++)
```



```

        printf(" ");
    space--;
    for (c = 1; c <= 2*k-1; c++)
        printf("*");
    printf("\n");
}

space = 1;
for (k = 1; k <= n - 1; k++)
{
    for (c = 1; c <= space; c++)
        printf(" ");

    space++;
    for (c = 1 ; c <= 2*(n-k)-1; c++)
        printf("*");
    printf("\n");
}
}

```

**Output**

Enter number of rows 5

```

      *
    ***
  *****
*****
*****
*****
  *****
    ***
      *

```

**34. Find the binary number for the given decimal number.**

```

#include<stdio.h>
int main(){
    int decimalNumber,remainder,quotient;
    int binaryNumber[100],i=1,j;

    printf("Enter any decimal number: ");
    scanf("%d",&decimalNumber);

    quotient = decimalNumber;

    while(quotient!=0){
        binaryNumber[i++]= quotient % 2;
        quotient = quotient / 2;
    }
}

```

```

printf("The Equivalent binary value for %d is ", decimalNumber);
for(j = i - 1 ; j > 0; j--)
printf("%d", binaryNumber[j]);

return 0;
}

```

**Output**

```

Enter any decimal number: 50
Equivalent binary value for 50 is 110010

```

**35. Write a C program to accept an array & swap element using pointers.**

```

#include <stdio.h>
void swap34(float *ptr1, float *ptr2);    //34 represents the elements 3 and 4 are to be
swapped.
void main()
{
float x[10];
int i, n;
printf("Enter how many Elements in an array ...\n");
scanf("%d", &n);
printf("Enter Elements of an array \n");
for (i = 0; i < n; i++)
{
scanf("%f", x + i);
}
/* Function call: Interchanging 3rd element by 4th */
swap34(x + 2, x + 3);
printf("\nResultant Array is ...\n");
for (i = 0; i < n; i++)
{
printf("X[%d] = %f\n", i, x[i]);
}
}
/* Function to swap the 3rd element with the 4th element in the array */
void swap34(float *ptr1, float *ptr2 )
{
float temp;
temp = *ptr1;
*ptr1 = *ptr2;
*ptr2 = temp;
}

```

**Output**

```

Enter how many elements in an array
5
Enter elements of an array

```

```
1
6
8
4
3
```

Resultant array is

```
X[0] = 1.000000
X[1] = 6.000000
X[2] = 4.000000
X[3] = 8.000000
X[4] = 3.000000
```

### 36. Write a C program to generate Armstrong number.

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

    printf("Enter the range value of a and b \n");
    scanf("%d%d", &a, &b);

    for (c = a; c <= b; c++) {
        if (check_armstrong(c) == 1)
            printf("\n The armstrong numbers are");
        printf("%d\n", c);
    }

    return 0;}

int check_armstrong(int n) {
    long long sum = 0, temp;
    int remainder, digits = 0;

    temp = n;

    while (temp != 0) {
        digits++;
        temp = temp/10;
    }

    temp = n;

    while (temp != 0) {
        remainder = temp%10;
        sum = sum + power(remainder, digits);
        temp = temp/10;
    }

    if (n == sum)
        return 1;
```

```
        else
            return 0;}
int power(int n, int r) {
    int c, p = 1;

    for (c = 1; c <= r; c++)
        p = p*n;

    return p;
}
```

**Output**

```
Enter the range value of a and b: 10 500
The Armstrong numbers are
153
370
371
407
```

**37. Find the sum of two numbers using pointers.**

```
#include <stdio.h>
int main(){
    int first, second, *p, *q, sum;

    printf("Enter the two values \n");
    scanf("%d%d", &first, &second);

    p = &first;
    q = &second;

    sum = *p + *q;

    printf("Sum of two numbers is = %d\n",sum);

    return 0;
}
```

**Output**

```
Enter the two values:
5 6
Sum of two numbers is = 11
```

**38. Bubble sort an array.**

```
#include <stdio.h>
int main(){
    int array[100], n, c, d, swap;

    printf("Enter the size of an array\n");
    scanf("%d", &n);
```

```

printf("Enter %d integers\n", n);
for (c = 0; c < n; c++)
    scanf("%d", &array[c]);
for (c = 0 ; c < ( n - 1 ); c++)
{
    for (d = 0 ; d < n - c - 1; d++)
    {
        if (array[d] > array[d+1])    /* For decreasing order use < */
        {
            swap      = array[d];
            array[d]   = array[d+1];
            array[d+1] = swap;
        }
    }
}
printf("Sorted list is:\n");
for ( c = 0 ; c < n ; c++ )
    printf("%d\n", array[c]);
return 0;
}

```

**Output**

```

Enter the size of an array: 5
Enter 5 integers
6
2
8
7
9
Sorted list is
2
6
7
8
9

```

**39. Merge the given arrays.**

```

#include <stdio.h>
void merge(int [], int, int [], int, int []);
int main() {
    int a[100], b[100], m, n, c, sorted[200];

    printf("Enter the values for FIRST array\n");
    scanf("%d", &m);

```

```
printf("Input %d integers\n", m);
for (c = 0; c < m; c++) {
    scanf("%d", &a[c]);
}

printf("Enter the values for SECOND array\n");
scanf("%d", &n);

printf("Input %d integers\n", n);
for (c = 0; c < n; c++) {
    scanf("%d", &b[c]);
}

merge(a, m, b, n, sorted);

printf("Sorted array is :\n");
for (c = 0; c < m + n; c++) {
    printf("%d\n", sorted[c]);
}

return 0;
}

void merge(int a[], int m, int b[], int n, int sorted[]) {
    int i, j, k;

    j = k = 0;

    for (i = 0; i < m + n && j < m && k < n; i++) {
        if (a[j] < b[k]) {
            sorted[i] = a[j];
            j++;
        }
        else {
            sorted[i] = b[k];
            k++;
        }
    }

    if (j == m) {
        for (; i < m + n; i++) {
            sorted[i] = b[k];
            k++;
            i++;
        }
    }
    else {
        for (; i < m + n; i++) {
            sorted[i] = a[j];
            j++;
            i++;
        }
    }
}
```

```
    }
  }
}
```

**Output**

```
Enter the values for FIRST array
3
Input 3 integers
7
9
15
Enter the values for SECOND array
3
Input 3 integers
2
6
78
Sorted array is
2
6
7
9
15
78
```

**40. Program for merging two files.**

```
#include <stdio.h>#include <stdlib.h>
int main()
{
    FILE *fs1, *fs2, *ft;
    char ch, file1[20], file2[20], file3[20];
    printf("Enter the 1st file name \n");
    gets(file1);
    printf("Enter the 2nd file name \n");
    gets(file2);
    printf("Enter the name of file to be merged\n");
    gets(file3);
    fs1 = fopen(file1,"r");
    fs2 = fopen(file2,"r");
    if( fs1 == NULL || fs2 == NULL )
    {
        perror("Error ");
        printf("Press any key to exit...\n");
        getch();
    }
}
```

```

        exit(EXIT_FAILURE);
    }
    ft = fopen(file3,"w");
    if( ft == NULL )
    {
        perror("Error ");
        printf("Press any key to exit...\n");
        exit(EXIT_FAILURE);
    }
    while( ( ch = fgetc(fs1) ) != EOF )
        fputc(ch,ft);
    while( ( ch = fgetc(fs2) ) != EOF )
        fputc(ch,ft);
    printf("Two files are merged into %s file successfully.\n",file3);
    fclose(fs1);
    fclose(fs2);
    fclose(ft);
    return 0;}

```

#### Output

```

Enter the 1st file name
file1
Enter the 2nd file name
file2
Enter the name of the file to be merged
file3
Two files were merged into file3 file successfully.

```

#### 41. Binary search in an array.

```

#include <stdio.h>
int main(){
    int c, first, last, middle,n, search, array[100];
    printf("Enter the size of an array\n");
    scanf("%d",&n);
    printf("\n Enter the values");
    for (c = 0; c < n; c++)
        scanf("%d",&array[c]);
    printf("Enter the value to be searched\n");
    scanf("%d", &search);
    first = 0;
    last = n - 1;
    middle = (first+last)/2;

```



```
while (first <= last) {
    if (array[middle] < search)
        first = middle + 1;
    else if (array[middle] == search) {
        printf("%d is found at location %d.\n", search, middle+1);
        break;
    }
    else
        last = middle - 1;
    middle = (first + last)/2;
}
if (first > last)
    printf("Not found! %d is not present in the list.\n", search);
return 0; }
```

**Output**

```
Enter the size of an array: 5
Enter the values
6
8
5
4
1
Enter the value to be searched: 5
5 is found at the location 3
```

**42. Linear search in an array.**

```
#include<stdio.h>
int main()
{
    int array[100],search,c,n;
    printf ("Enter the size of an array\n");
    scanf ("%d",&n);

    printf ("\n Enter the values of an array");
    for (c = 0; c < n; c++)
        scanf ("%d" ,&array[c]);
    printf("Enter the number to be searched");
    scanf ("%d",&search);
    for(c=0;c<n;c++)
        if(array[c]==search)
        {
            printf("%d is present at location %d.\n", search, c+1);
            break;
        }
}
```

```
    }  
    if (c == n)  
        printf("%d is not present in array.\n", search);  
    return 0;}
```

**Output**

```
Enter the size of an array: 5  
Enter the values of an array  
5  
4  
8  
9  
7  
Enter the number to be searched: 9  
9 is present at location 4.
```

**43. Find the reverse a number and check if it is a palindrome.**

```
#include <stdio.h>  
void main ()  
{  
    int num, temp, remainder, reverse = 0;  
    printf ("Enter a number \n");  
    scanf ("%d", &num);  
    temp = num;  
    while (num > 0)  
    {  
        remainder = num % 10;  
        reverse = reverse * 10 + remainder;  
        num /= 10;  
    }  
    printf ("Given number is = %d\n", temp);  
    if (temp == reverse)  
        printf ("The given Number is a palindrome \n");  
    else  
        printf ("The given Number is not a palindrome \n");  
}
```

**Output**

```
Enter a number  
121  
Given number is = 121  
The given number is a palindrome.
```

**44. Shared memory using unions.**

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

```

{
union number
{
int n1;
float n2;
};
union number x;

printf ("Enter the value of n1: ");
scanf ("%d", &x.n1);
printf ("Value of n1 is= %d", x.n1);
printf ("\nEnter the value of n2: ");
scanf ("%f", &x.n2);
printf ("Value of n2 is = %f\n", x.n2);
}

```

**Output**

```

Enter the value of n1: 5
Value of n1 is = 5
Enter the value of n2: 6.2
Value of n2 is = 6.200000

```

**45. Find the size of a union.**

```

#include <stdio.h>
void main()
{
union sample
{
int m;
float n;
char ch;
};
union sample u;

printf("The size of union is = %d\n", sizeof(u));
u.m = 25;
printf("%d %f %c\n", u.m, u.n, u.ch);
u.n = 0.2;
printf("%d %f %c\n", u.m, u.n, u.ch);
u.ch = 'p';
printf("%d %f %c\n", u.m, u.n, u.ch);
}

```

**Output**

```

The size of union is = 4
25 0.000000
1045220557 0.200000 = 1045220464 0.199999 p

```

**46. Find the sum of two complex numbers.**

```
#include <stdio.h>
struct complex
{
    int realpart, imaginary;
};
main()
{
    struct complex a, b, c;

    //Complex number is represented as a+ib; where a is a real part and b is an imaginary part.
    printf("Enter value of a and b .\n");
    printf("value of complex number a is = ");
    scanf("%d", &a.realpart);
    printf("value of complex number b is = ");
    scanf("%d", &a.imaginary);
    printf("Enter value of c and d .\n");
    printf("value of complex number c is = ");
    scanf("%d", &b.realpart);
    printf("value of complex number d is = ");
    scanf("%d", &b.imaginary);
    c.realpart = a.realpart + b.realpart;
    c.imaginary = a.imaginary + b.imaginary;
    if (c.imaginary >= 0)
        printf("The sum of complex numbers is = %d + %di\n", c.realpart, c.imaginary);
    else
        printf("The sum of complex numbers is = %d %di\n", c.realpart, c.imaginary);
    return 0;
}
```

**Output**

```
Enter value of a and b
Value of complex number a is = 5
Value of complex number b is = 6
Enter value of c and d
Value of complex number c is = 4
Value of complex number d is = 8
The sum of complex numbers is = 9 + 14i
```

**47. Find the length of the string using recursion.**

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

struct node
{
    char a;
    struct node *next;
```

```
};
void generate(struct node **);
int length(struct node*);
void delete(struct node **);

int main()
{
    struct node *head = NULL;
    int count;

    generate(&head);
    count = length(head);
    printf("The length of entered string is: %d\n", count);
    delete(&head);
    return 0;
}

void generate(struct node **head)
{
    /* for unknown number of nodes use num = rand() % 20; */
    int i;
    char str[50];
    printf("Enter the string: ");
    scanf("%s", str);
    struct node *temp;

    for (i = 0; str[i] != '\0'; i++)
    {
        {
            temp = (struct node *)malloc(sizeof(struct node));
            temp->a = str[i];
            if (*head == NULL)
            {
                *head = temp;
                (*head)->next = NULL;
            }
            else
            {
                {
                    temp->next = *head;
                    *head = temp;
                }
            }
        }
    }

    int length(struct node *head)
    {
        {
            if (head->next == NULL)
            {
                return 1;
            }
        }
    }
}
```

```
return (1 + length(head->next));
}

void delete(struct node **head)
{
    struct node *temp;
    while (*head != NULL)
    {
        temp = *head;
        *head = (*head)->next;
        free(temp);
    }
}
```

**Output**

```
Enter the string: structure
The length of string is: 9
```

**48. Reverse the string using recursion.**

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

void reverse(char [], int, int);
int main()
{
    char str1[20];
    int size;

    printf("Enter a string to reverse: ");
    scanf("%s", str1);
    size = strlen(str1);
    reverse(str1, 0, size - 1);
    printf("After reversing a string : %s\n", str1);
    return 0;
}

void reverse(char str1[], int index, int size)
{
    char temp;
    temp = str1[index];
    str1[index] = str1[size - index];
    str1[size - index] = temp;
    if (index == size / 2)
    {
        return;
    }
    reverse(str1, index + 1, size);
}
```

**Output**

Enter a string to reverse: goodmorning  
After reversing a string: gniromdoog

**49. Find the product of two numbers using addition without using recursion.**

```
#include <stdio.h>
int product(int, int);
int main()
{
    int a, b, result;
    printf("Enter two numbers ");
    scanf("%d%d", &a, &b);
    result = product(a, b);
    printf("Product of %d and %d is %d\n", a, b, result);
    return 0;
}
int product(int a, int b)
{
    int temp = 0;
    while (b != 0)
    {
        temp += a;
        b--;
    }
    return temp;
}
```

**Output**

Enter two numbers: 8 6  
Product of 8 and 6 is 48

**50. Find out the roots of quadratic equation.**

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
void main()
{
    float a, b, c, root1, root2;
    float realp, imagp, disc;

    printf("Enter the values of a, b and c \n");
    scanf("%f %f %f", &a, &b, &c);

    /* If a = 0, it is not a quadratic equation */
    if (a == 0 || b == 0 || c == 0)
    {
```

```
printf("Roots cannot be determined ,Enter the valid number \n");
exit(1);
}
else
{
disc = b * b - 4.0 * a * c;
if (disc < 0)
{
printf("Imaginary Roots\n");
realp = -b / (2.0 * a) ;
imagp = sqrt(abs(disc)) / (2.0 * a);
printf("Root1 = %f +i %f\n", realp, imagp);
printf("Root2 = %f -i %f\n", realp, imagp);
}
else if (disc == 0)
{
printf("Roots are real and equal\n");
root1 = -b / (2.0 * a);
root2 = root1;
printf("Root1 = %f\n", root1);
printf("Root2 = %f\n", root2);
}
else if (disc > 0 )
{
printf("Roots are real and distinct \n");
root1 = (-b + sqrt(disc)) / (2.0 * a);
root2 = (-b - sqrt(disc)) / (2.0 * a);
printf("Root1 = %f \n", root1);
printf("Root2 = %f \n", root2);
}
}
}
```

**Output**

```
Enter the values of a, b and c
5 10 2
Roots are real and distinct
Root1 = -0.225403
Root2 = -1.774597
```