

Name : Sharma Anish Ashok

Roll No : 11

SAP ID : 60003220045

Branch : Information Technology

Div : D

Batch : IT1

Experiment No. 11:

Write C program to find GCD of two integers by using recursive function. Write algorithm and draw flowchart for the same.

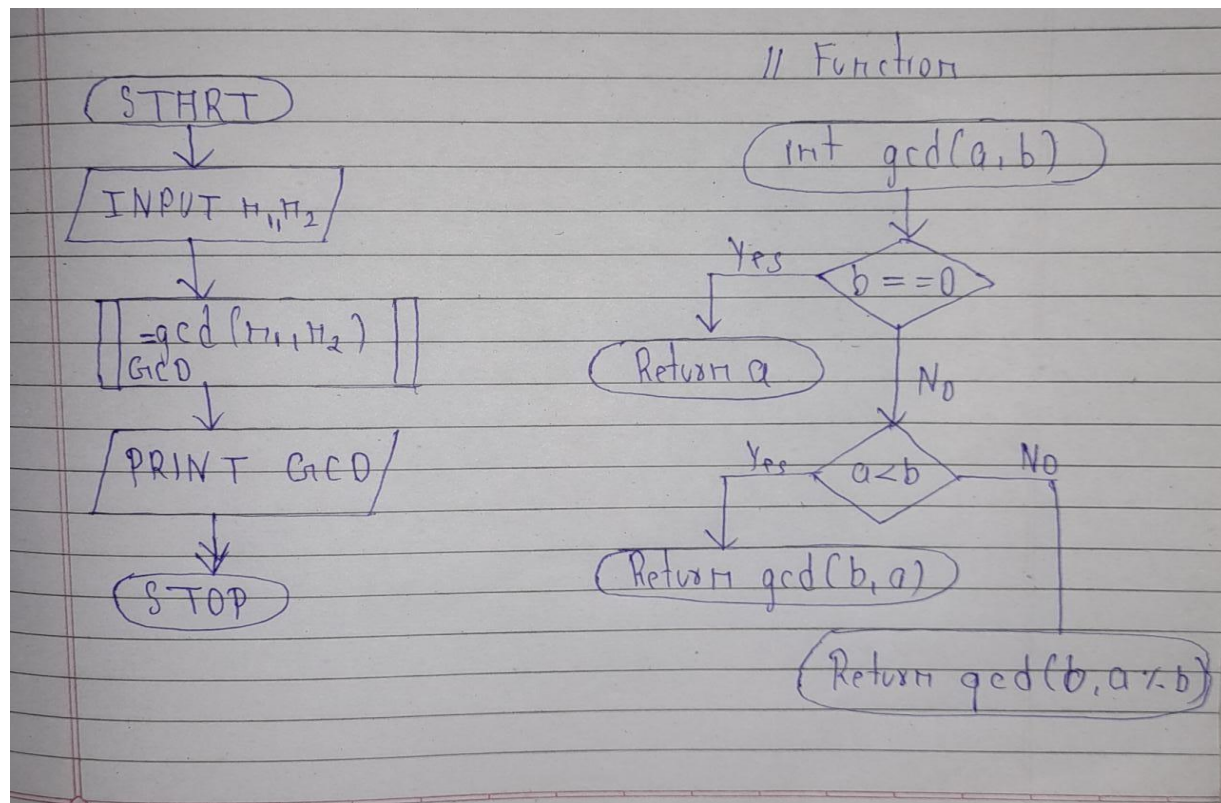
Aim : To find GCD of two integers by using recursive function

Algorithm :

```
1. START
2. Read  $H_1, H_2$ 
3. Calling a function  $gcd(H_1, H_2)$  by value and store
   into GCD and Display GCD
4. STOP

// Function gcd()
1. START function definition as  $gcd(a, b)$ 
2. if  $b == 0$  then return a value
3. Otherwise if  $a < b$  then return  $gcd(b, a)$ 
4. Otherwise return  $gcd(a, b)$ 
5. STOP
```

Flowchart :



Code :

```
#include<stdio.h>

#include<conio.h>

int gcd(int a,int b)
{
    if(b==0)
        return a;
    else{
        if(a<b)
            return gcd(b,a);
        else
            return gcd(b,a%b);
    }
}

void main()
{
    int n1,n2,GCD;
    //clrscr();
    printf("Enter your two number :\n");
    scanf("%d %d",&n1,&n2);
    GCD=gcd(n1,n2);
    printf("GCD of %d and %d is %d",n1,n2,GCD);
    getch();
}
```

Output:

```
Enter your two number :  
5  
10  
GCD of 5 and 10 is 5|
```

```
Enter your two number :  
27  
91  
GCD of 27 and 91 is 1|
```

Conclusion :

We learn the recursive function through this example .And also the importance of recursive function in a program of GCD.

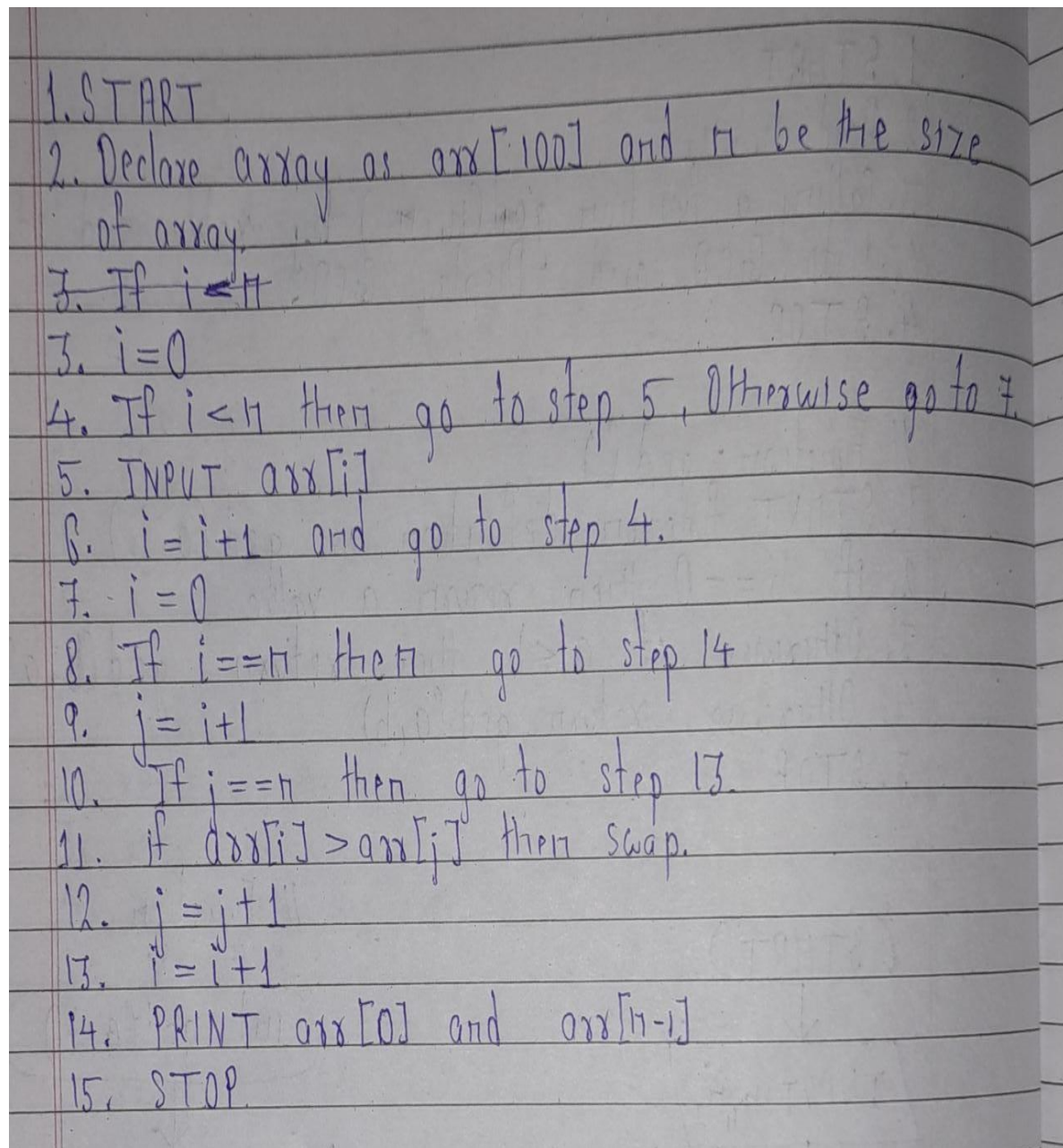
Experiment 12 :

Write a C program to find both the largest and smallest number in a list of integers. Write algorithm and draw flowchart for the same.

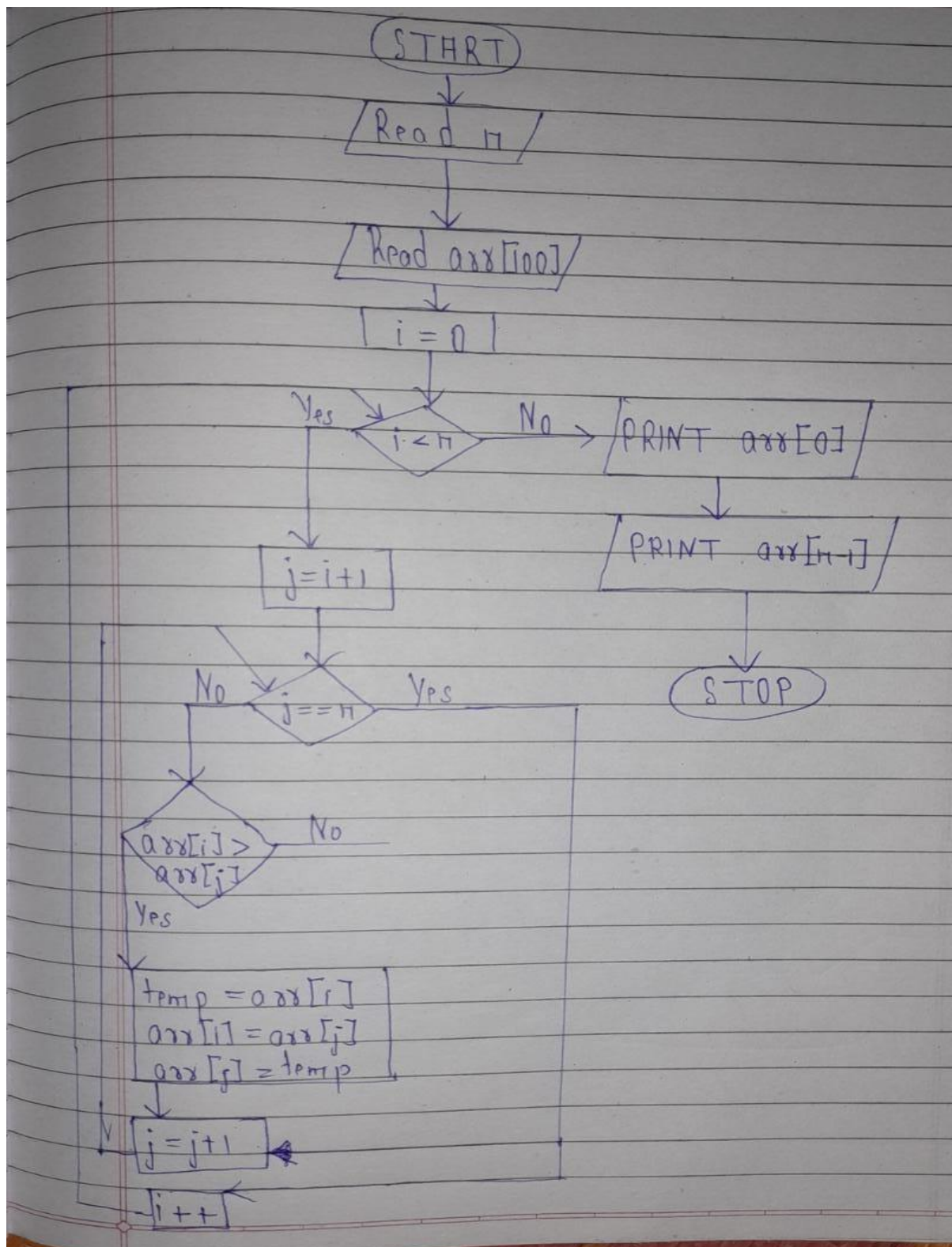
Aim :

To find largest and smallest number in a list of integers.

Algorithm :



Flowchart :



Code :

```
#include<stdio.h>

#include<conio.h>

void main()
{
    int i,j,arr[90],n,temp=0;
    //clrscr();
    printf("Enter array size:\n");
    scanf("%d",&n);
    printf("Enter array element:\n");
    for(i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    printf("\nPrinting array:\n");
    for(i=0;i<n;i++)
    {
        printf("%d ",arr[i]);
    }
    for(i=0;i<n;i++)
    {
        for(j=i+1;j<n;j++)
        {
            if(arr[i]>arr[j]){
                temp=arr[i];
                arr[i]=arr[j];
```



```

        arr[j]=temp;
    }
}

printf("\nSmallest element from list is :%d",arr[0]);
printf("\nLargest element from list is %d",arr[n-1]);
getch();
}

```

Output :

```

Enter array size:
7
Enter array element:
56
33
98
76
12
44
90

Printing array:
56 33 98 76 12 44 90
Smallest element from list is :12
Largest element from list is 98|

```

```

Enter array size:
5
Enter array element:
2
5
8
1
9

Printing array:
2 5 8 1 9
Smallest element from list is :1
Largest element from list is 9|

```

Conclusion :

Due to this program we aware of array and it's important in day to day life. Array is a collection of element of same data type .

Experiment 13 :

Develop, implement and execute a C program that reads two matrices A (m x n) and B (p x q) and Compute product of matrices A and B. Read matrix A and matrix B in row major order and in column major order respectively. Print both the input matrices and resultant matrix with suitable headings and output should be in matrix format only.

Aim :

Study of matrix and 2D array.

Code :

```
#include<stdio.h>

#include<conio.h>

void input_print_matrix(int matrix[10][10],int row,int col)
{
    int i,j;
    printf("\nEnter matrix element :\n");
    for(i=0;i<row;i++)
    {
        for(j=0;j<col;j++)
        {
            scanf("%d",&matrix[i][j]);
        }
    }
    printf("\nPrinting matrix :\n");
    for(i=0;i<row;i++)
    {
        for(j=0;j<col;j++)
        {
```

```

        printf("%d ",matrix[i][j]);
    }
    printf("\n");
}
}

void main()
{
    int i,j,k,row1,col1,row2,col2,sum,m1[10][10],m2[10][10],m3[10][10];
    //clrscr();
    printf("\nEnter row and column of matrix 1:\n");
    scanf("%d %d",&row1,&col1);
    printf("\nEnter row and column of matrix 2:\n");
    scanf("%d %d",&row2,&col2);
    if(col1==row2)
    {
        input_print_matrix(m1,row1,col1);
        //print_matrix(m1,row1,col1);
        input_print_matrix(m2,row2,col2);
        //print_matrix(m2,row2,col2);
        printf("\nMultiplication of two matrix is:\n");
        for(i=0;i<row1;i++)
        {
            for(j=0;j<col2;j++)
            {
                sum=0;
                for(k=0;k<row1;k++)

```

```

        {
            sum=sum+m1[i][k]+m2[k][j];
        }
        m3[i][j]=sum;
    }
}
for(i=0;i<row1;i++)
{
    for(j=0;j<col2;j++)
    {
        printf("%d ",m3[i][j]);
    }
    printf("\n");
}
}
else
    printf("\nMultiplication not possible ...");
getch();
}

```

Output :

```

Enter row and column of matrix 1:
3
2

Enter row and column of matrix 2:
3
3

Multiplication not possible ...|

```

Enter row and column of matrix 1:

2

2

Enter row and column of matrix 2:

2

2

Enter matrix element :

1

2

3

4

Printing matrix :

1 2

3 4

Enter matrix element :

12

3

4

4

Printing matrix :

12 3

4 4

Multiplication of two matrix is:

19 10

23 14

```
Enter row and column of matrix 1:  
3  
3
```

```
Enter row and column of matrix 2:  
3  
2
```

```
Enter matrix element :  
1  
2  
3  
4  
5  
6  
7  
8  
9
```

```
Printing matrix :  
1 2 3  
4 5 6  
7 8 9
```

```
Enter matrix element :  
1  
2  
3  
4  
5  
6
```

```
Printing matrix :  
1 2  
3 4  
5 6
```

```
Multiplication of two matrix is:  
15 18  
24 27  
33 36
```

Conclusion :

Through this program we learn multiplication of matrix along with 2D array .

Experiment No. 14

Write a Program for deletion of an element from the specified location from Array.

Aim :

Deletion of an element from array

Code :

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i,j=0,n,pos,arr[100];
    //clrscr();
    printf("Enter your array size:\n");
    scanf("%d",&n);
    printf("\nEnter your element is array:\n");
    for(i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    for(i=0;i<n;i++)
    {
        printf("%d ",arr[i]);
    }
    printf("\nEnter your position to want to delete that element from array:\n");
    scanf("%d",&pos);
    if(pos>=n)
        printf("Out of range !\nEnter valid position of element");
    else
    {
        printf("\nYour element is deleted :%d",arr[pos]);
        for(i=0;i<n;i++)
        {
            if(arr[pos]!=arr[i])
            {
                arr[j]=arr[i];
```

```

        j++;
    }
}
printf("\nPrinting Array after deletion process:\n");
for(i=0;i<n-1;i++)
{
    printf("%d ",arr[i]);
}
}
getch();
}

```

Output :

```

Enter your array size:
6

Enter your element is array:
1
3
8
7
0
5
1 3 8 7 0 5
Enter your position to want to delete that element from array:
4

Your element is deleted :0
Printing Array after deletion process:
1 3 8 7 5 |

```

```

Enter your array size:
4

Enter your element is array:
1
2
3
4
1 2 3 4
Enter your position to want to delete that element from array:
5
Out of range !
Enter valid position of element|

```



```
Enter your array size:
10

Enter your element is array:
22
99
77
88
34
54
67
32
80
97
22 99 77 88 34 54 67 32 80 97
Enter your position to want to delete that element from array:
9

Your element is deleted :97
Printing Array after deletion process:
22 99 77 88 34 54 67 32 80 |
```

Conclusion :

We studied and practice complex array through this program. How to delete an element from array .

Experiment 15 :

Write a C program using user defined functions to determine whether the given string is palindrome or not.

Aim :

Study of palindrome using user defined function.

Code :

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n,rem,store,rev=0;
    //clrscr();
    printf("Enter your number :\n");
    scanf("%d",&n);
    store=n;
    while(n>0){
        rem=n%10;
        rev=rev*10+rem;
        n/=10;
    }
    if(store==rev)
        printf("\nThe given number is palindrome");
    else
        printf("\nThe given number is not palindrome");

}
```

Output :

```
Enter your number :  
1234
```

```
The given number is not palindrome
```

```
Enter your number :  
12121
```

```
The given number is palindrome
```

Conclusion :

We learn how to check whether a number is palindrome or not by using loop concept .