

Name : Ayush Jain
SAP ID : 60004200132
Branch : Computer Engineering
Div : J1

CP lab experiments (11-20)

Aim:

- (1) Write a Program to find nC_r using function. Write algorithm and draw flowchart for the same.

Algorithm:

Step 1 : Start

Step 2 : Print "Enter the values of n and r"

Step 3 : Input n, r

Step 4 : ${}^nC_r = \text{call factorial (arguments: } n \text{)}$

$[\text{call factorial (arguments: } r \text{)}]^* [\text{call factorial (arguments: } n-r \text{)}]$

Step 5 : Print nC_r

Step 6 : Stop

Step 1 : Start

Step 2 : $i=1$, fact = 1

Step 3 : If $i > r$, then goto step 7

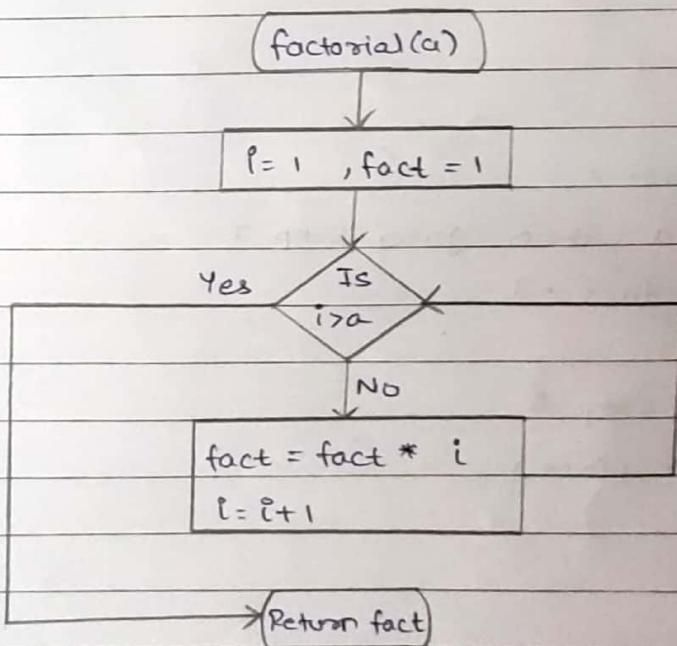
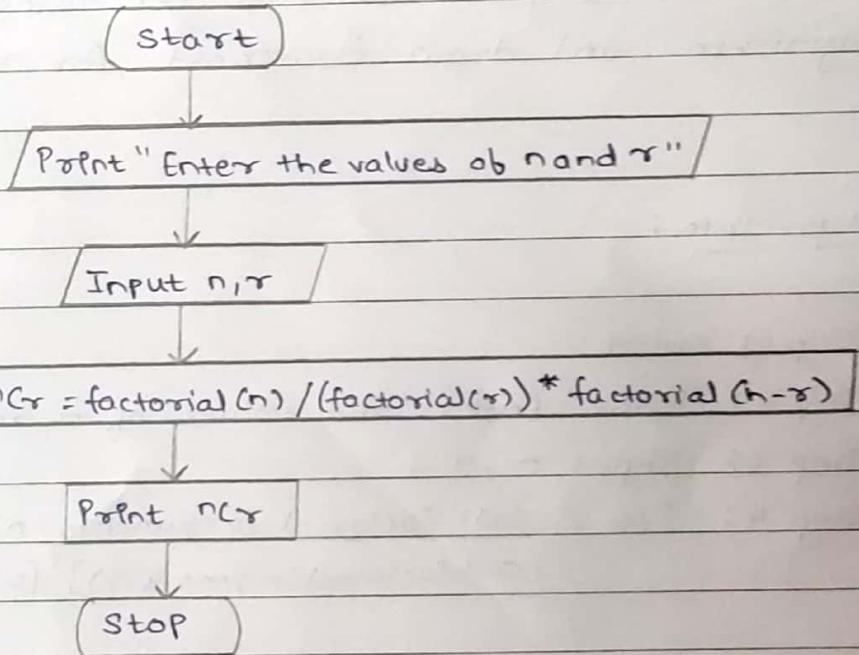
Step 4 : fact = fact * i

Step 5 : $i = i + 1$

Step 6 : Goto step 3

Step 7 : Return (fact)

Flowchart:



Program:

```
#include <stdio.h>
void main()
{
    int n, r, nCr;
    int fact(int no);
    printf ("Enter the values of n and r : ");
    scanf ("%d %d", &n, &r);
    nCr = fact(n) / [fact(r)* fact(n-r)];
    printf ("nCr = %d", nCr);
}

int fact (int no)
{
    int i, ans;
    for (i=1; ans=1; i<=no; i++)
    {
        ans = ans * i;
    }
    return ans;
}
```

Output :

Enter the values of n and r : 10 2
nCr = 45

12.) Aim: Write a program which will accept n and r and calculate $nCr = n! / r!(n-r)!$ using recursive functions.

Program:

```
#include <stdio.h>
void main()
{
    int n, r, nCr;
    int fact(int no);
    printf("Enter the values of n and r: ");
    scanf("%d %d", &n, &r);
    nCr = fact(n) / (fact(r) * fact(n-r));
    printf ("nCr = %d", nCr);
}

int fact (int no)
{
    if (no == 1)
        return 1;
    else
        return (no * fact(no-1));
}
```

Output:

Enter the values of n and r: 10 3
 $nCr = 120$

13) Aim: Write a program to initialize an automatic and static variable and increment it in the function. Call this function thrice and print the value of the variable every time after incrementing.

Program:

```
#include <stdio.h>
int main()
{
    int i;
    void check();
    for(i=0;i<3;i++)
    {
        check();
    }
    return 0;
}

void check()
{
    auto int a = 5;
    static int b = 10;
    a++;
    b++;
    printf("The values of automatic variable after incrementation is : %d \n", a);
    printf("The values of static variable after incrementation is : %d \n", b);
}
```

Output :

The value of automatic variable after incrementation is : 6

The value of static variable after incrementation is : 11

The value of automatic variable after incrementation is : 6

The value of static variable after incrementation is : 12

The value of automatic variable after incrementation is : 6

The value of static variable after incrementation is : 13

14) Aim: Write a program to implement bubble sorting algorithm for sorting numbers in ascending order. Write algorithm and draw flowchart for the same.

Algorithm:

Step 1 : Start

Step 2 : Input n

Step 3 : i = 0

Step 4 : If $i < n$

 else go to step 8

Step 5 : Input $a[i]$

Step 6 : $i = i + 1$

Step 7 : Goto Step 4

Step 8 : $i = 0$

Step 9 : If $i < n$

 Else goto step 21

Step 10 : $j = 0$

Step 11 : If $j < n - i - 1$

 Else goto step 19

Step 12 : If $a[j] > a[j+1]$

 Else goto step 17

Step 13 : temp = $a[j]$

Step 14 : $a[j] = a[j+1]$

Step 15 : $a[j+1] = \text{temp}$

Step 16 : Go to step 17

Step 17 : $j = j + 1$

Step 18 : Go to step 11

Step 19 : $i = i + 1$

Step 20 : Go to step 9

Step 21 : $i = 0$

Step 22 : If $i < n$

Else go to Step 27

Step 23 : Print $a[i]$

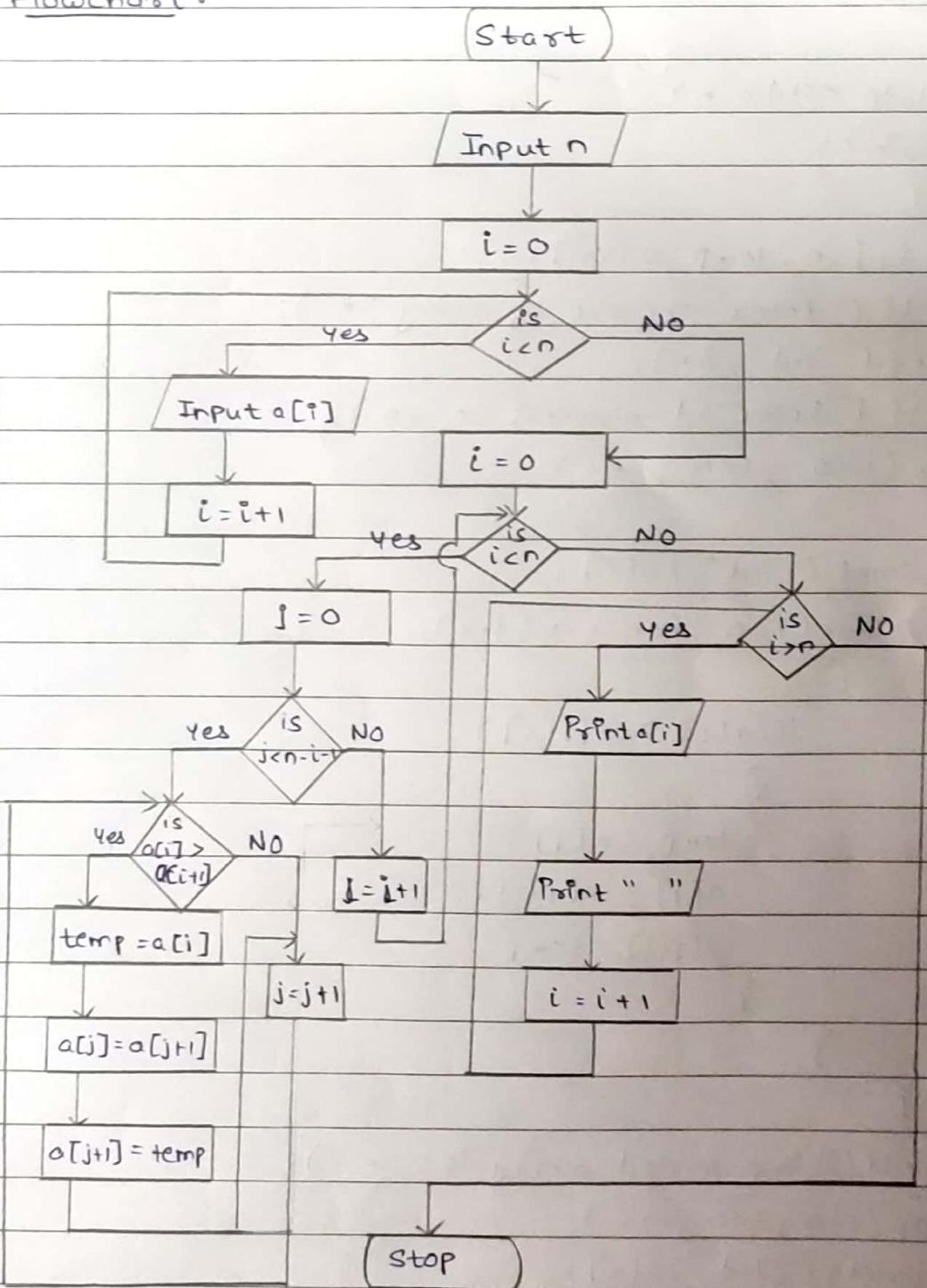
Step 24 : Print "

Step 25 : $i = i + 1$

Step 26 : Go to step 22

Step 27 : Stop

Flowchart:



Program:

```
# include <stdio.h>
int main()
{
    int i, j, n, temp, a[100];
    printf ("Enter the size of array \n");
    scanf ("%d", &n);
    printf ("Enter %d elements \n", n);
    for (i=0 ; i<n ; i++)
    {
        scanf ("%d", &a[i]);
        for (j=0 ; j<n-i-1 ; j++)
        {
            if (a[j] > a[j+1])
            {
                temp = a[j];
                a[j] = a[j+1];
                a[j+1] = temp;
            }
        }
    }
    printf ("The sorted array is: \n");
    for (i=0 ; i<n ; i++)
        printf ("%d ", a[i]);
    return 0;
}
```

Output:

Enter the size of the array

6

Enter 6 elements

7

3

9

5

2

8

The sorted array is:

2 3 5 7 8 9

15) Aim: Write a program to check whether string is pallindrome or not

Program:

```
#include <stdio.h>
#include <string.h>
int main ()
{
    char s[50], s1[50];
    int i, j, x=0, y=0;
    printf("Enter a string \n");
    gets(s);
    for (i=0; s[i]!='\0'; i++)
        x++;
    for (j=0; j=x-1; j >=0; i++, j--)
        s1[i] = s[j];
    s1[x] = '\0';
    for (i=0; i<x; i++)
    {
        if (s[i] == s1[i])
            y++;
        if (y == x)
            printf ("Pallindrome");
        else
            printf ("Not a Pallindrome");
    }
}
```

Output:

Enter a String: racecar
: Pallindrome .

16) Aim: Write a program to count blank spaces, digits, vowels and consonants in the string.

Program:

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

int main ()
{
    char s[50];
    int i, a, x=0, y=0, z=0, b=0;
    printf("Enter a string: ");
    gets(s);
    for (i=0; s[i] != '\0'; i++)
        x++;
    for (i=0; i<x; i++)
    {
        if (s[i] == 'A' || s[i] == 'E' || s[i] == 'I' || s[i] == 'O' || s[i] == 'U'
            || s[i] == 'a' || s[i] == 'e' || s[i] == 'i' || s[i] == 'o' || s[i] == 'u')
            y++;
    }
    for (i=0; i<x; i++)
    {
        if (s[i] == '1' || s[i] == '2' || s[i] == '3' || s[i] == '4' || s[i] == '5' ||
            s[i] == '6' || s[i] == '7' || s[i] == '8' || s[i] == '9' || s[i] == '0')
            z++;
    }
}
```

```

for(i=0; i<x; i++)
{
    if(s[i] == ' ')
        b++;
}
a = x - y - z - b;
printf ("Number of vowels are %.d \n", y);
printf ("Number of consonants are %.d \n", a);
printf ("Number of numbers are %.d \n", z);
printf ("Number of spaces are %.d \n", b);

return 0;
}

```

Output :

Enter a string :

My name is Ayush123

Number of vowels are 5

Number of consonants are 8

Number of numbers are 3

Number of spaces are 3

(7) Aim: Write a Program to multiply two matrices using a function.

Program:

```
#include <stdio.h>
int main ()
{
    int a[10][10], b[10][10], m, n, p, i, j, k;
    printf("Enter m, n and p \n");
    scanf("%d %d %d", &m, &n, &p);
    printf("Enter the values of matrix A \n");
    for(i=0 ; i<m ; i++)
        for(j=0 ; j<n ; j++)
    {
        scanf ("%d", &a[i][j]);
    }
    printf("Enter the value of matrix B \n");
    for(i=0 ; i<n ; i++)
        for(j=0 ; j<p ; j++)
    {
        scanf ("%d", &b[i][j]);
    }
    int c[m][p]
    for(i=0 ; i<m ; i++)
    {
        for(j=0 ; j<p ; j++)
    {
```

```

c[i][j] = 0 ;
for (k=0 ; k < n ; k++)
{
    c[i][j] = c[i][j] + a[i][k] * b[k][j] ;
}
for (i=0 ; i < m ; i++)
{
    for (j=0 ; j < p ; j++)
    {
        printf ("%d \t", c[i][j]);
    }
    printf ("\n");
}
return 0;
}

```

Output :

Enter m, n and p

3 2 3

Enter the values of matrix A

1

2

3

4

5

6

Enter the value of matrix B

1

2

3

4

5

6

(
9 12 15
19 26 33
29 40 51

18) Aim: Define a structure called cricket that will describe the following information - Player name , country name , best score , batting average . Develop a program that will store information of 25 cricket players around the world using this structure. Also display names of these cricketers in descending order with respect to their batting average.

Program:

```
# include <stdio.h>
Struct cricket
{
    char player_name [20];
    char country_name [20];
    int best_score ;
    float average;
};

typedef struct cricket ct;

int main()
{
    ct C[25]; temp;
    int i, j, n;
    printf(" Enter number of players \n");
    scanf(" %d ", &n);
    for (i=0 ; i<n ; i++)
    {
        printf(" For player : %d \n ", i+1);
        printf(" Enter player name : ");
    }
}
```

```

scanf ("%s", &c[i].player-name);
printf("Enter country name : ");
scanf ("%s", &c[i].country-name);
printf("Enter best score : ");
scanf ("%d", &c[i].best-score);
printf("Enter Average : ");
scanf ("%f", &c[i].average);

for(i=0 ; i<n ; i++)
{
    for(j=0 ; j<n-i-1 ; j++)
    {
        if(c[j+1].average > c[i].average)
        {
            temp = c[j];
            c[i] = c[j+1];
            c[j+1] = temp;
        }
    }
}

printf("The sorted structure according to average is \n");
for(i=0 ; i<n ; i++)
{
    printf("Player name : %s \n", c[i].player-name);
    printf("Country name : %s \n", c[i].country-name);
    printf("Best score : %d \n", c[i].best-score);
    printf("Average score : %0.2f \n", c[i].average);
}
return 0;
}

```

Output:

Enter number of player: 3

For player: 1

Enter player name: Root

Enter country name: England

Enter best score: 133

Enter average: 77.80

For player: 2

Enter player name: Virat Kohli

Enter country name: India

Enter best score: 183

Enter average: 99.04

For player: 3

Enter player name: AB De Villiers

Enter country name: South Africa

Enter best score: 136

Enter average: 82.77

The sorted structure according to average is:

Player name: Virat Kohli

Country name: India

Best score: 183

Average score: 99.05

Player name: AB de Villiers

Country name: South Africa

Best score: 136

Average score: 82.77

Player name: Root

Country name: England

Best score: 133

Average score: 77.80



19) Aim: Write a program to swap two numbers using a function.
Pass the values to be swapped to this function using call-by-value method and call-by-reference method.

Program:

```
#include <stdio.h>
int main()
{
    int i, j;
    void call-by-value(int, int);
    void call-by-reference(int*, int*);
    i = 10;
    j = 20;
    printf(" Initial value of i is: %.d \n", i);
    printf(" Initial value of j is: %.d \n", j);
    call-by-value(i, j);
    printf(" Swap by call by value method : \n");
    printf(" Value of i is : %.d \n", i);
    printf(" Value of j is : %.d \n", j);
    call-by-reference(&i, &j);
    printf(" Swap by call by reference method : \n");
    printf(" Value of i is : %.d \n", i);
    printf(" Value of j is : %.d \n", j);

    return 0;
}
```

void call-by-value (int a, int b)

{

 int temp;

 temp = a;

 a = b;

 b = temp;

}

void call-by-reference (int *a, int *b)

{

 int temp;

 temp = *a;

 *a = *b;

 *b = temp;

}

Output :

Initial value of i is : 10

Initial value of j is : 20

Swap by call by value method:

Value of i is : 10

Value of j is : 20

Swap by call by reference method:

Value of i is : 20

Value of j is : 10

Q. 20) Aim: Write a program to accept a set of 10 numbers and print the number using pointer.

Program:

```
#include <stdio.h>
int main()
{
    int i, a[10], * p ;
    printf("Enter 10 elements \n");
    for (i=0 ; i<10 ; i++)
        scanf ("%d", &a[i]);
    p=a;
    printf (" Displaying array elements using pointers \n");
    for (i=0 ; i<10 ; i++)
        printf ("%d", *(p+i));
    return 0;
}
```

Output:

Enter 10 elements

7 4 9 11 2 3 8 14 1 15

Displaying array elements using pointer

7 4 9 11 2 3 8 14 1 15