

WEEK 11:

Objective: Explore the Functions, sub-routines, scope and extent of variables, doing some experiments by parameter passing using call by value. Basic methods of numerical integration

Suggested Experiments/Activities:

Tutorial 11: Functions, call by value, scope and extent,

Lab 11: Simple functions using call by value, solving differential equations using Eulers theorem.

i) Write a C function to calculate NCR value.

ii) Write a C function to find the length of a string.

iii) Write a C function to transpose of a matrix.

iv) Write a C function to demonstrate numerical integration of differential equations using Euler's method

i) Write a C function to calculate NCR value.

```
#include <stdio.h>
long factorial(int);
long find_ncr(int, int);
long find_npr(int, int);

int main()
{
    int n, r;
    long ncr, npr;
    printf("Enter the value of n and r:");
    scanf("%d%d", &n, &r);
    ncr = find_ncr(n, r);
    npr = find_npr(n, r);
    printf("%dC%d = %ld", n, r, ncr);
    printf("\n%dP%d = %ld", n, r, npr);
    return 0;
}

long find_ncr(int n, int r) {
    long result;
    result = factorial(n)/(factorial(r)*factorial(n-r));
    return result;
}
```

```

long find_npr(int n, int r)
{
    long result;
    result = factorial(n)/factorial(n-r);
    return result;
}
long factorial(int n)
{
    int i;
    long f = 1;
    if(n==0)
        return f;
    else
    {
        for (i = 2; i <= n; i++)
            f = f*i;
        return f;
    }
}

```

Input/Output:

1: Enter the value of n and r: 5 2

5C2 = 10

5P2 = 20

2: Enter the value of n and r: 10 5

10C5 = 252

10P5 = 30240

ii) Write a C function to find the length of a string.

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

```

/* Function Prototype */
int mystrlen(char str[30]);

```

```

/* Main Function */
int main()

```

```

{
char str[30];
int i, len;
printf("Enter string:\n");
gets(str);
len = mystrlen(str); /* Function Call */
printf("Length of given string is: %d", len);
return 0;
}

```

```

/* Function Definition */
int mystrlen(char str[30])
{
int i, len=0;
for(i=0;str[i]!='\0';i++)
{
len++;
}
return(len);
}

```

Input/Output:

Enter string: c programming
Length of given string is: 13

iii)Write a C function to transpose of a matrix

```

#include<stdio.h>
void transpose(int a[10][10], int n);
int main()
{
int i,j,n;
int a[10][10];
printf("Enter order of matrix:\n");
scanf("%d", &n);
fflush(stdin);
printf("Enter matrix elements:\n");

```

```

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

```

```

transpose(a,n);
printf("\nDisplay transpose of matrix:\n");
for(i=0;i< n;i++)
{
    for(j=0;j< n;j++)
    {
        printf("%d\t",a[i][j]);
    }
    printf("\n");
}

```

```

return 0;
}

```

```

void transpose(int a[10][10], int n)
{
    int i,j;
    int tmp;
    for(i=0;i<n-1;i++)
    {   for(j=i+1;j< n;j++)
        {   tmp = a[i][j];
            a[i][j] = a[j][i];
            a[j][i] = tmp;
        }
    }
}

```

Input/Output ::

Enter order of matrix:

2 2

Enter matrix elements:

a[0][0]=1

a[0][1]=2

a[1][0]=3

a[1][1]=4

Display transpose of matrix:

1 3

2 4

iv) Write a C function to demonstrate numerical integration of differential equations using Euler's method

Note: A simple differential equation $dy/dx = x + y$ and uses Euler's method to numerically solve it for a specific range of x values.

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

```
#define f(x,y) x+y
```

```
void eulerIntegration(float ,float,float,float,int);
```

```
int main()
```

```
{
```

```
float x0, y0, xn, h;
```

```
int i, n;
```

```
printf("Enter Initial Condition\n");
```

```
printf("x0 = ");
```

```
scanf("%f", &x0);
```

```
printf("y0 = ");
```

```
scanf("%f", &y0);
```

```
printf("Enter calculation point xn = ");
```

```
scanf("%f", &xn);
```

```
printf("Enter number of steps: ");
```

```
scanf("%d", &n);
```

```
h = (xn-x0)/n; /* Calculating step size (h) */
```

```
eulerIntegration(x0, y0, h, xn,n);
```

```
return 0;
```

```

}
void eulerIntegration(float x0,float y0,float h,float xn,int n)
{ int i;
  float yn, slope;
  printf("\nx0\ty0\tslope\ty0\n");
  printf("-----\n");
  for(i=0; i < n; i++)
  {   slope = f(x0, y0);
      yn = y0 + h * slope;
      printf("%.4f\t%.4f\t%.4f\t%.4f\n",x0,y0,slope,yn);
      y0 = yn;
      x0 = x0+h;
  }
  printf("\nValue of y at x = %.2f is %.3f",xn, yn);
}

```

Input/Output:

Enter Initial Condition

x0 = 0

y0 = 1

Enter calculation point xn = 1

Enter number of steps: 10

x0	y0	slope	yn
0.0000	1.0000	1.0000	1.1000
0.1000	1.1000	1.2000	1.2200
0.2000	1.2200	1.4200	1.3620
0.3000	1.3620	1.6620	1.5282
0.4000	1.5282	1.9282	1.7210
0.5000	1.7210	2.2210	1.9431
0.6000	1.9431	2.5431	2.1974
0.7000	2.1974	2.8974	2.4872
0.8000	2.4872	3.2872	2.8159
0.9000	2.8159	3.7159	3.1875

Value of y at x = 1.00 is 3.187