

Lab 07

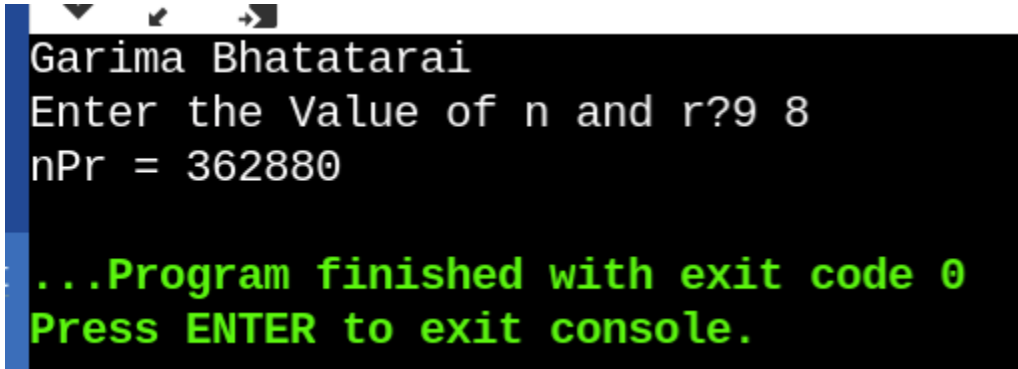
Title: To find permutation and combination of given number

A permutation is an act of arranging the objects or numbers in order.

Combinations are the way of selecting the objects or numbers from a group of objects or collection,

1. Program to generate permutation of a given number

```
#include<stdio.h>
int main ()
{
    int n, r, per, fact1, fact2, number, i;
    printf("Enter the Value of n and r?");
    scanf("%d %d", &n, &r);
    fact1 = n;
    for (int i = n - 1; i >= 1; i--)
    {
        fact1 = fact1 * i;
    }
    number = n - r;
    fact2 = number;
    for (i = number - 1; i >= 1; i--)
    {
        fact2 = fact2 * i;
    }
    per = fact1 / fact2;
    printf("nPr = %d", per);
    return 0;
}
```



Garima Bhatatarai

Enter the Value of n and r?9 8

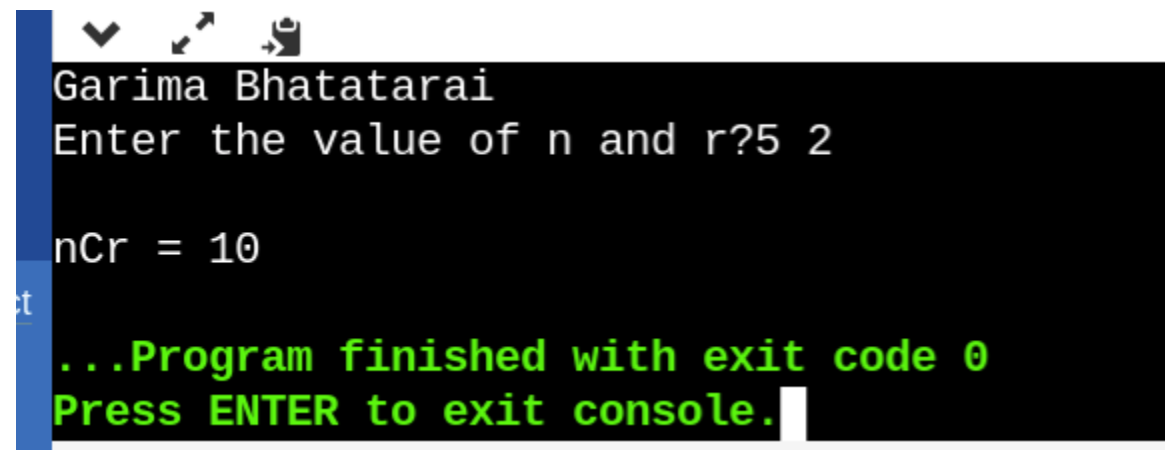
nPr = 362880

...Program finished with exit code 0

Press ENTER to exit console.

2. Program to generate combination of a given number

```
#include <stdio.h>
int fact(int z);
int main()
{
    int n, r, nCr;
    printf("Garima Bhatatarai\n");
    printf("Enter the value of n and r?");
    scanf("%d %d", &n, &r);
    nCr = fact(n) / (fact(r) * fact(n - r));
    printf("\nnCr = %d", nCr);
}
int fact(int z)
{
    int f = 1, i;
    if (z == 0)
        return (f);
    else
    {
        for (i = 1; i <= z; i++)
        {
            f = f * i;
        }
    }
    return (f);
}
```



```
Garima Bhatatarai
Enter the value of n and r?5 2

nCr = 10

...Program finished with exit code 0
Press ENTER to exit console.
```

Lab 8:

Title : Program to implement Dijkstra's algorithm to find shortest path

Dijkstra's algorithm is an algorithm for finding the shortest paths between nodes in a graph, which may represent, for example, road networks.

```
#include <stdio.h>
#define infinity 999
```

```
void dij(int n,int v,int cost[10][10],int dist[])
{
    int i,u,count,w,flag[10],min;
    for(i=1;i<=n;i++)
        flag[i]=0,dist[i]=cost[v][i];
    count=2;
    while(count<=n)
    {
        min=99;
        for(w=1;w<=n;w++)
            if(dist[w]<min && !flag[w])
                min=dist[w],u=w;
        flag[u]=1;
        Count++;
        for(w=1;w<=n;w++)
            if((dist[u]+cost[u][w]<dist[w]) && !flag[w])
                dist[w]=dist[u]+cost[u][w];
    }
}
```

```
int main()
{
    int n,v,i,j,cost[10][10],dist[10];

    printf("\n Enter the number of nodes:");
    scanf("%d",&n);
    printf("\n Enter the cost matrix:\n");
    for(i=1;i<=n;i++)
        for(j=1;j<=n;j++)
        {
            scanf("%d",&cost[i][j]);
            if(cost[i][j]==0)
                cost[i][j]=infinity;
        }
    printf("\n Enter the source matrix:");
    scanf("%d",&v);
    dij(n,v,cost,dist);
    printf("\n Shortest path:\n");
}
```

```
for(i=1;i<=n;i++)  
    if(i!=v)  
        printf("%d->%d,cost=%d\n",v,i,dist[i]);  
}
```

Garima Bhattarai

Enter the number of nodes:2

Enter the cost matrix:

8

6

4

2

Enter the source matrix:8



Shortest path:

8->1,cost=6

8->2,cost=14090