

1. // Write a program to generate permutations

```
#include<stdio.h>

int fact(int n){
    if(n == 0 )
        return 1;
    else
        return n * fact(n - 1);
}

int main(){
    int n, r;

    printf("Enter the total number of items: ");
    scanf("%d", &n);

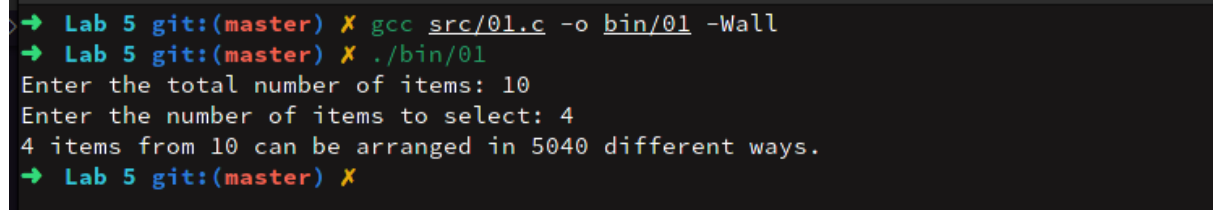
    printf("Enter the number of items to select: ");
    scanf("%d", &r);

    int permutation;
    permutation = fact(n) / fact(n - r);

    printf("%d items from %d can be arranged in %d different
ways.\n", r, n, permutation);

}
```

OUTPUT:



```
> → Lab 5 git:(master) X gcc src/01.c -o bin/01 -Wall
→ Lab 5 git:(master) X ./bin/01
Enter the total number of items: 10
Enter the number of items to select: 4
4 items from 10 can be arranged in 5040 different ways.
→ Lab 5 git:(master) X
```

2. //Write a program to generate combination

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

    if(n == 0 )
        return 1;
    else
        return n * fact(n - 1);
}

int main(){
    int n, r;

    printf("Enter the total number of items: ");
    scanf("%d", &n);

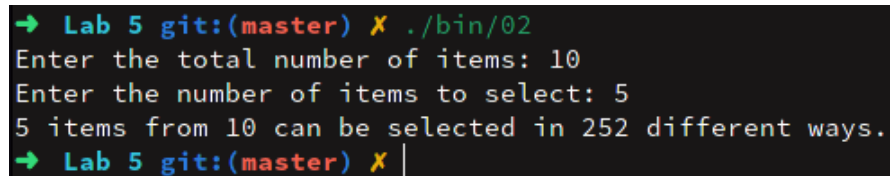
    printf("Enter the number of items to select: ");
    scanf("%d", &r);

    int combination;
    combination = fact(n) / (fact(r)*fact(n - r));

    printf("%d items from %d can be selected in %d different
ways.\n", r, n, combination);

}
```

OUTPUT:



```
→ Lab 5 git:(master) X ./bin/02
Enter the total number of items: 10
Enter the number of items to select: 5
5 items from 10 can be selected in 252 different ways.
→ Lab 5 git:(master) X |
```