

Name:T.Deekshitha

Reg: 192011256

1.fibonacci using recursion

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

```
void printFibonacci(int n){
```

```
    static int n1=0,n2=1,n3;
```

```
    if(n>0){
```

```
        n3 = n1 + n2;
```

```
        n1 = n2;
```

```
        n2 = n3;
```

```
        printf("%d ",n3);
```

```
        printFibonacci(n-1);
```

```
    }
```

```
}
```

```
int main(){
```

```
    int n;
```

```
    printf("Enter the number of elements: ");
```

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

```
    printf("Fibonacci Series: ");
```

```
    printf("%d %d ",0,1);  
    printFibonacci(n-2);  
    return 0;  
}
```

Output:

```
Enter the number of elements: 7  
Fibonacci Series: 0 1 1 2 3 5 8  
Process returned 0 (0x0)   execution time : 9.514 s  
Press any key to continue.
```

2.Armstrong number

```
#include<stdio.h>  
  
int main()  
{  
    int n,r,sum=0,temp;  
    printf("enter the number:");  
    scanf("%d",&n);  
    temp=n;  
    while(n>0)
```

```

{
    r=n%10;
    sum=sum+(r*r*r);
    n=n/10;
}
if(temp==sum)
    printf("armstrong number");
else
    printf("not armstrong number");
return 0;
}

```

Output:



```

C:\Users\91911\OneDrive\Documents\Exams\p1\armstrong number.exe
Enter the number:153
armstrong number
Process returned 0 (0x0)   execution time : 16.462 s
Press any key to continue.

```

3.GCD of two numbers

```
#include <stdio.h>

int main()
{
    int n1, n2, i, gcd;

    printf("Enter two integers: ");
    scanf("%d %d", &n1, &n2);

    for(i=1; i <= n1 && i <= n2; ++i)
    {
        // Checks if i is factor of both integers
        if(n1%i==0 && n2%i==0)
            gcd = i;
    }

    printf("G.C.D of %d and %d is %d", n1, n2,
gcd);

    return 0;
}
```

Output:



```
"C:\Users\91911\OneDrive\Documents\Exams\ptr\gcd of 2 num.exe"
Enter two integers: 10 4
G.C.D of 10 and 4 is 2
Process returned 0 (0x0) execution time : 6.212 s
Press any key to continue.
```

4.Largest number in an array

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

```
int main() {
```

```
    int n;
```

```
    double arr[100];
```

```
    printf("Enter the number of elements (1 to 100):
");
```

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

```
    for (int i = 0; i < n; ++i) {
```

```
        printf("Enter number%d: ", i + 1);
```

```
        scanf("%lf", &arr[i]);
```

```
}
```

```
// storing the largest number to arr[0]
```

```
for (int i = 1; i < n; ++i) {
```

```
    if (arr[0] < arr[i]) {
```

```
        arr[0] = arr[i];
```

```
    }
```


```
}
```

```
printf("Largest element = %.2lf", arr[0]);
```

```
return 0;
```

```
}
```

Output:

A screenshot of a C++ program execution in a terminal window. The program prompts the user to enter the number of elements, followed by four numbers. It then outputs the largest element and the execution time.

```
Enter the number of elements ): 4
Enter number1: 23
Enter number2: 45
Enter number3: 12
Enter number4: 4
Largest element = 45.00
Process returned 0 (0x0)   execution time : 11.225 s
Press any key to continue.
```

5.Factorial of a number using recursion

```
#include<stdio.h>

long int multiplyNumbers(int n);

int main() {
    int n;
    printf("Enter a positive integer: ");
    scanf("%d",&n);
    printf("Factorial of %d = %ld", n,
multiplyNumbers(n));
    return 0;
}

long int multiplyNumbers(int n) {
    if (n>=1)
        return n*multiplyNumbers(n-1);
    else
        return 1;
}
```

Output:



6.To check the given number is prime or not

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

```
int main() {
```

```
    int n, i, flag = 0;
```

```
    printf("Enter a positive integer: ");
```

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

```
    if (n == 0 || n == 1)
```

```
        flag = 1;
```

```
    for (i = 2; i <= n / 2; ++i) {
```

```
        if (n % i == 0) {
```

```
            flag = 1;
```

```
            break;
```



```

    }
}

if (flag == 0)
    printf("%d is a prime number.", n);
else
    printf("%d is not a prime number.", n);

return 0;
}

```

Output:



```

C:\Users\9191\OneDrive\Documents\Exams\ptr\prime or not.exe
Enter a positive integer: 3
3 is a prime number.
Process returned 0 (0x0) execution time : 4.046 s
Press any key to continue.

```

7. Selection sort

```

#include <stdio.h>

void swap(int *xp, int *yp)
{
    int temp = *xp;

```

```
    *xp = *yp;  
    *yp = temp;  
}
```

```
void selectionSort(int arr[], int n)  
{  
    int i, j, min_idx;  
    for (i = 0; i < n-1; i++)  
    {  
        min_idx = i;  
        for (j = i+1; j < n; j++)  
            if (arr[j] < arr[min_idx])  
                min_idx = j;  
        if(min_idx != i)  
            swap(&arr[min_idx], &arr[i]);  
    }  
}
```

```
void printArray(int arr[], int size)  
{
```

```
int i;
for (i=0; i < size; i++)
    printf("%d ", arr[i]);
printf("\n");
}

int main()
{
    int arr[] = {64, 25, 12, 22, 45};
    int n = sizeof(arr)/sizeof(arr[0]);
    selectionSort(arr, n);
    printf("Sorted array: \n");
    printArray(arr, n);
    return 0;
}
```

Output:



```
"C:\Users\91911\OneDrive\Documents\Exams\ptr\selection sort.exe"
Sorted array:
12 22 25 45 64
Process returned 0 (0x0)   execution time : 4.255 s
Press any key to continue.
```