Name : Manzoor Mohd

Class : AIML

Roll no : 1604-21-748-025

Assignment - 4

1. Write a program to add, subtract and multiply two complex numbers using structures to function.

Ans

```
#include<stdio.h>
typedef struct complex {
        float real;
        float imag;
} complex;

complex add(complex n1, complex n2);
complex sub(complex n1, complex n2);
complex mul(complex n1, complex n2);

int main() {
        int choice;
        complex n1, n2, result;
        printf("For 1st complex number \n");
        printf("Enter the real and imaginary parts: ");
        scanf("%f %f", &n1.real, &n1.imag);
```

```
complex sub(complex n1, complex n2) {
        complex temp;
        temp.real = n1.real - n2.real;
        temp.imag = n1.imag - n2.imag;
        return (temp);
}

complex mul(complex n1, complex n2) {
        complex temp;
        temp.real = (n1.real * n2.real) - (n1.imag * n2.imag);
       temp.imag = (n1.real * n2.imag) + (n1.imag * n2.real);
       return (temp);
}
```

Write a structure to store the name, account number and balance of customers and store their information.

```
struct customer
{
    int account_no;
    char name[80];
    int balance;
}s[20];
// to store
for (i = 0; i <= n; i++)  /h is the limit of
scanf("%d %c %d", &s[i].account_no, &s[i].name, &s[i].balance);
```

```
    printf("\nFor 2nd complex number \n");
    printf("Enter the real and imaginary parts: ");
    scanf("%f %f", &n2.real, &n2.imag);
    printf ("Select (1 - 3)  \n");
    printf ("1 - Addition\n");
    printf ("2 - Substraction\n");
    printf ("3 - Multiplication\n");
    scanf ("%d", &choice);
     if (choice == 1)
    result = add(n1, n2);
    else if (choice == 2)
    result = sub(n1, n2);
    else if (choice == 3)
    result = mul(n1, n2);
    printf("Reultant complex number = %.1f + %.1fi", result.real,
     result.imag);
   return 0;
  }
complex add(complex n1, complex n2) {
        complex temp;
        temp.real = n1.real + n2.real;
        temp.imag = n1.imag + n2.imag;
        return(temp);
```

3. Write a program in C to get the largest element of an array using recursion.

```
#include<stdio.h>
#define N 5

void biggest(int *num, int n, int big)
{
    if(n < 0)
    printf("Biggest element is %d\n", big);
    else
    {
        if(*num > big)
        big = *num;
        biggest(++num, --n, big);
    }
}
int main()
{
    int a[N], i;
    printf("Enter %d integer numbers\n", N);
    for(i = 0; i < N; i++)
    scanf("%d", &a[i]);
    biggest(a, N - 1, a[0]);
    return 0;
}
```

5 Write a program in C to reverse a string using recursion.

```
#include<stdio.h>
void swap(char *x, char *y)
{
    char temp = *x;
    *x = *y;
    *y = temp;
}
void reverse(char *str, int k)
{
      static int i = 0;
      if (*(str + k) == '\0') {
      return;
      }
      reverse(str, k + 1);
      if (i <= k) {
      swap(&str[i++], &str[k]);
      }
}
int main()
{
      char str[] = "Techie Delight";
      reverse(str, 0);
      printf("Reverse of the given string is %s", str);
      return 0;
}
```

7. Write a program in C to reverse a string using recursion.

```
#include<stdio.h>
void swap(char *x, char *y)
{
    char temp = *x;
    *x = *y;
    *y = temp;
}
void reverse(char *str, int k)
{
    static int i = 0;
    if (*(str + k) == '\0') {
    return;
    }
    reverse(str, k + 1);
    if (i <= k) {
    swap(&str[i++], &str[k]);
    }
}
int main()
{
    char str[] = "Techie Delight";
    reverse(str, 0);
    printf("Reverse of the given string is %s", str);
    return 0;
```