PHASE 1 DAY 6

classwork

1. Pattern printing using nested loops

```
#include <stdio.h>

int main()
{
    int i = 1, j;

    while (i <= 5)
    {
        j = 1;
        while (j <= i)
        {
            printf("* ");
            j++;
        }
        printf("\n");
        i++;
    }
    return 0;
}
```

Output:

```
*
* *
* * *
* * * *
* * * * *
```

2.
Pyramid pattern

3.Multiplication table

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

int main()
```

```
{
    int i=1,j;
    do
    {
     j=1;
     do
     {
     printf("%d * %d =%d ",i,j,i*j);
     j++;
     }while(j<=10);
     printf("\n");
     i++;

    }
    while(i<=10);

    return 0;
}
```

4.Sum of N natural no.s using for loop

```
#include<stdio.h>
int main()
{
    int num,sum=0;
    printf("Enter the number:");
    scanf("%d",&num);

    for(int i=0;i<=num;i++)
    {
        sum +=i;
    }
    printf("=%d",sum);
}
```

5.Reverse number

```
#include<stdio.h>
int main()
{
    int num,rev=0,rem;
    printf("Enter the number:");
    scanf("%d",&num);
    for(;num!=0;num/=10)
    {
        rem=num%10;
        rev=rev*10 +rem;

    }
    printf("Reversed=%d",rev);
}
```

6.fibonacci series

```
#include<stdio.h>
int main()
{
    int num;
    printf("Enter the number:");
    scanf("%d",&num);

    int first=0,second=1,next;
    printf("Fibonacci series upto %d : %d %d",num,first,second);
    for(;next<=num;next++)
    {
       next=first + second;
       printf(" %d ",next);
       first=second;
       second=next;

    }
```

7.Pascal triangle

```
#include <stdio.h>

int main()
{
    int n;

    printf("Enter the number of rows for Pascal's Triangle: ");
    scanf("%d", &n);


    for (int i = 0; i < n; i++)
    {
        int k = 1;

        for (int j = 0; j < n - i - 1; j++)
        {
            printf(" ");
        }

        for (int j = 0; j <= i; j++)
        {
            if (j == 0 || j == i)
            {
                printf("1 ");
            }
            else
            {
                k = k * (i - j + 1) / j;
                printf("%d ", k);
            }
        }
        printf("\n");
    }

    return 0;
}
```

8.

Requirements

•In this challenge, you are going to create a "Guess the Number" C program

•Your program will generate a random number from 0 to 20

You will then ask the user to guess it

•User should only be able to enter numbers from 0-20

•The program will indicate to the user if each guess is too high or too low

The player wins the game if they can guess the number within five tries

Sample Output

```
This is a guessing game.

I have chosen a number between 0 and 20 which you must guess.

You have 5 tries left.

Enter a guess: 12

Sorry, 12 is wrong. My number is less than that.

You have 4 tries left.

Enter a guess: B Sorry, 8 is wrong. My number is less than that.

You have 3 tries left. Enter a guess: 4

Sorry, 4 is wrong. My number is less than that.

You have 2 tries left.

Enter a guess: 2

Congratulations. You guessed it!
```

Answer:

```
#include <stdio.h>
#include<stdlib.h>
#include<time.h>
int main()
{

   srand(time(NULL));
   int guess=rand()%21; //limit 0-20
   int num;

   //int guess=6;


   printf("This is a guessing game.\nI have chosen a number between 0 and 20 which you must
guess.");

   int chances=5;
```

```
    while(chances>=1)
    {
    printf("\nYou have %d tries left",chances);
    printf("\nEnter a guess:");
    scanf("%d",&num);

    if(num>=0 && num<=20)
    {

    if(num>guess)
    {
        printf("Sorry,%d is wrong.My number is less than that",num);
        chances--;
    }
    else if(num<guess)
    {
        printf("Sorry,%d is wrong.My number is greater than that",num);
        chances--;
    }
    else
    {
       printf("Congragulations!You guessed it ");

    }

    }
    else
    printf("Enter a number between 0 and 20");
    }
    printf("\nChances over");
    printf("\nThe correct number is %d",guess);
    return 0;
}
```

9.Program to find sum of all even numbers entered by the user

```
#include <stdio.h>


int main()
{
    int num;
    int sum=0;
    printf("Enter up to 20 integers:(Enter -1 to stop)");

    for(int i=0;i<=20;i++)

    {
     scanf("%d",&num);

     if(num ==-1)
       {
           break;
       }

       if(0!=num%2)
       {
       continue;
       }

        if(0==num%2 && num>0)
       {
          sum+=num;
       }

    }

   printf("sum of even numbers:%d",sum);

     return 0;
}
```

10.

Problem Statement 1: Banking System Simulation

Description: Create a simple banking system simulation that allows users to create an account, deposit money, withdraw money, and check their balance. The program should handle multiple accounts and provide a menu-driven interface.

Requirements:

1. Use appropriate data types for account balance (e.g., float for monetary values) and user input (e.g., int for account numbers).

2. Implement a structure to hold account details (account number, account holder name, balance).

3. Use control statements to navigate through the menu options:

- **i. Create Account**
- **ii. Deposit Money**
- **iii. Withdraw Money**
- **iv. Check Balance**

4. Ensure that the withdrawal does not exceed the available balance and handle invalid inputs gracefully.

Example Input/Output:

```
 Welcome to the Banking System
 1. Create Account
 2. Deposit Money
 3. Withdraw Money
 4. Check Balance
 5. Exit
 Choose an option: 1
 Enter account holder name: John Doe
 Account created successfully! Account Number: 1001


Choose an option: 2
 Enter account number: 1001
 Enter amount to deposit: 500
 Deposit successful! New Balance: 500.0


Choose an option: 3
 Enter account number: 1001
 Enter amount to withdraw: 200
 Withdrawal successful! New Balance: 300.0



Choose an option: 4
 Enter account number: 1001
 Current Balance: 300.0



Choose an option: 5
 Exiting the system.
```

Answer:

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

int main()
{
   char name[50];
   int acc_no,option;
   float deposit,withdraw,acc_balance=0;


   while(1)
   {


   printf("\n\nWelcome to the Banking system\n1.Create Account\n2.Deposit
Money\n3.Withdraw money\n4.Check Balance\n5.Exit");
   printf("\nChoose an option: ");
   scanf("%d",&option);

   switch(option)
   {
     case 1:
     {
        printf("\nEnter account Holder Name:");
        scanf("%s",name);
        printf("\nAccount created successfully!\nAccount number:1001");
        break;
     }
     case 2:
```

```
  {
     printf("\nEnter the Account number:");
     scanf("%d",&acc_no);

     printf("\nEnter amount to deposit:");
     scanf("%f",&deposit);
     acc_balance=acc_balance+deposit;
     printf("\nDeposit successful!\nNew balance:Rs.%.2f",acc_balance);
     break;
  }
  case 3:
  {
     printf("\nEnter Account Number:");
     scanf("%d",&acc_no);
     printf("\nEnter amount to withdraw:");
     scanf("%f",&withdraw);
     if(withdraw>acc_balance)
     {
      printf("\nInsufficient balance");
      break;
     }
     else
     {
     acc_balance=acc_balance-withdraw;
     printf("\nWithdraw successful!\nNew balance:Rs.%.2f",acc_balance);
     }
      break;
  }
  case 4:
  {
     printf("\nEnter account number:");
     scanf("%d",&acc_no);
     printf("\n Current Balance=Rs.%.2f",acc_balance);
     break;
  }
  case 5:
  {
     printf("\nExiting System!");
     exit(0);
  }
  default:
  printf("invalid input");

}
```

```
    }
    return 0;
}
```

11.

Problem Statement 4: Weather Data Analysis

Description: Write a program that collects daily temperature data for a month and analyzes it to find the average temperature, the highest temperature, the lowest temperature, and how many days were above average.

Requirements:

1. **Use appropriate data types (float for temperatures and int for days).**
2. **Store temperature data in an array.**
3. **Use control statements to calculate:**
    - i. **Average Temperature of the month.**
    - ii. **Highest Temperature recorded.**
    - iii. **Lowest Temperature recorded.**
    - iv. **Count of days with temperatures above average.**
4. **Handle cases where no data is entered.**

Example Input/Output:

```
 Enter temperatures for each day of the month (30 days):
 Day 1 temperature: 72.5
 Day 2 temperature: 68.0
 ...
 Day 30 temperature: 75.0


Average Temperature of Month: XX.X
 Highest Temperature Recorded: YY.Y
 Lowest Temperature Recorded: ZZ.Z
 Number of Days Above Average Temperature: N
```

Answer:

```
#include <stdio.h>
int main()
{
    float month[30];
    float sum=0,average;
    int count_above_avg=0;
     printf("Enter temperatures for each day of the month(30 Days):");
       for(int i=0;i<30;i++)
    {
        printf("\nDay %d temperature:",i+1);
        scanf("%f",&month[i]);
        sum += month[i];
    }
    float lowest_temp=month[0];
    float highest_temp=month[0];
    average = sum/30;
    for (int i = 1; i < 30; i++)
    {
        if (month[i] > highest_temp)
        {
            highest_temp = month[i];
        }
        if (month[i] < lowest_temp)
        {
            lowest_temp = month[i];

        }

        if(month[i] > average)

        {

            count_above_avg++;

        }

    }

    printf("\nAverage Temperature of Month:%.1f",average);

    printf("\nHighest Temperature of Recorded:%.1f",highest_temp);

    printf("\nLowest Temperature of Recorded:%.1f",lowest_temp);

    printf("\nNumber of Days Above Average Temperature:%d",count_above_avg);

}
```

12.Problem Statement : Inventory Management System

Description: Create an inventory management system that allows users to manage products in a store. Users should be able to add new products, update existing product quantities, delete products, and view inventory details.

Requirements:
1. Use appropriate data types for product details (e.g., char arrays for product names, int for quantities, float for prices).

2. Implement a structure to hold product information.

3. Use control statements for menu-driven operations:

i. Add Product
ii. Update Product Quantity
iii. Delete Product
iv. View All Products in Inventory

4. Ensure that the program handles invalid inputs and displays appropriate error messages.

Example Input/Output:

```
Inventory Management System
 1. Add Product
 2. Update Product Quantity
 3. Delete Product
 4. View All Products in Inventory
 5. Exit


Choose an option: 1
 Enter product name: Widget A
 Enter product quantity: 50
 Enter product price: 19.99


Choose an option: 4
 Product Name: Widget A, Quantity: 50, Price: $19.99


Choose an option: 5
 Exiting the system.
```

Answer:

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

```
int main()

{

    char product[30];

    int quantity =0,option;

    float price;

    while(1)

    {

        printf("1.Add Product \n");

        printf("2.update product quantity \n");

        printf("3.delete product \n");

        printf("4.view all products in inventory\n");

        printf("5.Exit \n");

        printf("choose an option");

        scanf("%d",&option);

        switch (option)

        {

        case 1:

            printf("enter product name \n");

            scanf(" %[^\n]%*c", product);

            printf("enter product quantity \n");

            scanf("%d",&quantity);

            printf("enter productprice \n");

            scanf("%f",&price);
            break;

        case 2:

            printf("updating product quantity :%d \n",quantity);

            break;

        case 3:

            printf("deleting product \n");

            if(quantity=0)

            {

                printf("empty");

                break;

            }

            quantity=quantity-1;

            break;

            case 4:

            printf("product name %c \n",product);

            printf("quantity : %d \n",quantity);

            printf("price is %f \n",price);

            break;

        case 5:

            printf("exiting system \n");

            return 0;

        default:

            break;

        } } }
```