

DAY 6

1. //nested while loop

//create a multiplication table from 1 to 10

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

```
int main()
```

```
{
```

```
    int num=1;
```

```
    int i;
```

```
    while(num<=10)
```

```
    {
```

```
        i=1;
```

```
        while(i<=10)
```

```
        {
```

```
            printf("%d * %d =%d\t",num,i,num*i);
```

```
            i++;
```

```
        }
```

```
        printf("\n");
```

```
        num++;
```

```
    }
```

```
}
```

```
PS D:\learning c\output> & .\day6_2.exe'
1 * 1 =1      1 * 2 =2      1 * 3 =3      1 * 4 =4      1 * 5 =5      1 * 6 =6      1 * 7 =7      1 * 8 =8      1 * 9 =9      1 * 10 =10
2 * 1 =2      2 * 2 =4      2 * 3 =6      2 * 4 =8      2 * 5 =10     2 * 6 =12     2 * 7 =14     2 * 8 =16     2 * 9 =18     2 * 10 =20
3 * 1 =3      3 * 2 =6      3 * 3 =9      3 * 4 =12     3 * 5 =15     3 * 6 =18     3 * 7 =21     3 * 8 =24     3 * 9 =27     3 * 10 =30
4 * 1 =4      4 * 2 =8      4 * 3 =12     4 * 4 =16     4 * 5 =20     4 * 6 =24     4 * 7 =28     4 * 8 =32     4 * 9 =36     4 * 10 =40
5 * 1 =5      5 * 2 =10     5 * 3 =15     5 * 4 =20     5 * 5 =25     5 * 6 =30     5 * 7 =35     5 * 8 =40     5 * 9 =45     5 * 10 =50
6 * 1 =6      6 * 2 =12     6 * 3 =18     6 * 4 =24     6 * 5 =30     6 * 6 =36     6 * 7 =42     6 * 8 =48     6 * 9 =54     6 * 10 =60
7 * 1 =7      7 * 2 =14     7 * 3 =21     7 * 4 =28     7 * 5 =35     7 * 6 =42     7 * 7 =49     7 * 8 =56     7 * 9 =63     7 * 10 =70
8 * 1 =8      8 * 2 =16     8 * 3 =24     8 * 4 =32     8 * 5 =40     8 * 6 =48     8 * 7 =56     8 * 8 =64     8 * 9 =72     8 * 10 =80
9 * 1 =9      9 * 2 =18     9 * 3 =27     9 * 4 =36     9 * 5 =45     9 * 6 =54     9 * 7 =63     9 * 8 =72     9 * 9 =81     9 * 10 =90
```

2. //pattern printing

```
*
```

```
**
```

```
***
```

```
****
```

```
*****
```

```

#include<stdio.h>

int main()
{
    int i=1,j;
    while(i<=5)
    {
        j=1;
        while(j<=i)
        {
            printf("* ");
            j++;
        }
        printf("\n");
        i++;
    }
}

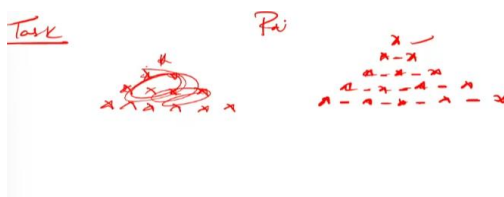
```

```

PS D:\learning c\output> & .\'day6_3.exe\'
*
* *
* * *
* * * *
* * * * *

```

3.



//pattern printing

```

#include <stdio.h>

```

```
int main() {  
    int rows = 5;  
    int i = 1, j, k;  
  
    while (i <= rows) {  
  
        k = rows - i;  
        while (k > 0) {  
            printf("\t");  
            k--;  
        }  
  
        j = 1;  
        while (j <= (2 * i - 1)) {  
            if (j % 2 != 0) {  
                printf("*\t");  
            } else {  
                printf("\t");  
            }  
            j++;  
        }  
  
        printf("\n");  
        i++;  
    }  
  
    return 0;  
}
```

```
PS D:\learning c\output> cd d:\learning c\output
PS D:\learning c\output> & .\'day6_4.exe'

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

4. //WAP to print numbers between 1 to 10 using do while

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

```
int main()
```

 $\{$

```
int i=1;
```

do{

```
printf("%d \n",i);
```

```
    i++;
```

```
}while(i<=10);
```

}

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

5. //WAP to print multiplication table using do while

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

```
int main()
```

 $\{$

```
int i,num=1;
```

do{

```
i=1;
```

```
do
{
printf("%d * %d = %d \n",num,i,num*i);

i++;

}while(i<=10);

num++;

}while(num<=10);
}
```

1 * 1 = 1

1 * 2 = 2

1 * 3 = 3

1 * 4 = 4

1 * 5 = 5

1 * 6 = 6

1 * 7 = 7

1 * 8 = 8

1 * 9 = 9

1 * 10 = 10

2 * 1 = 2

2 * 2 = 4

2 * 3 = 6

2 * 4 = 8

2 * 5 = 10

2 * 6 = 12

2 * 7 = 14

2 * 8 = 16

2 * 9 = 18

2 * 10 = 20

3 * 1 = 3

3 * 2 = 6

3 * 3 = 9

$$3 * 4 = 12$$

$$3 * 5 = 15$$

$$3 * 6 = 18$$

$$3 * 7 = 21$$

$$3 * 8 = 24$$

$$3 * 9 = 27$$

$$3 * 10 = 30$$

$$4 * 1 = 4$$

$$4 * 2 = 8$$

$$4 * 3 = 12$$

$$4 * 4 = 16$$

$$4 * 5 = 20$$

$$4 * 6 = 24$$

$$4 * 7 = 28$$

$$4 * 8 = 32$$

$$4 * 9 = 36$$

$$4 * 10 = 40$$

$$5 * 1 = 5$$

$$5 * 2 = 10$$

$$5 * 3 = 15$$

$$5 * 4 = 20$$

$$5 * 5 = 25$$

$$5 * 6 = 30$$

$$5 * 7 = 35$$

$$5 * 8 = 40$$

$$5 * 9 = 45$$

$$5 * 10 = 50$$

$$6 * 1 = 6$$

$$6 * 2 = 12$$

$$6 * 3 = 18$$

$$6 * 4 = 24$$

$$6 * 5 = 30$$

$$6 * 6 = 36$$

$$6 * 7 = 42$$

$$6 * 8 = 48$$

$$6 * 9 = 54$$

$$6 * 10 = 60$$

$$7 * 1 = 7$$

$$7 * 2 = 14$$

$$7 * 3 = 21$$

$$7 * 4 = 28$$

$$7 * 5 = 35$$

$$7 * 6 = 42$$

$$7 * 7 = 49$$

$$7 * 8 = 56$$

$$7 * 9 = 63$$

$$7 * 10 = 70$$

$$8 * 1 = 8$$

$$8 * 2 = 16$$

$$8 * 3 = 24$$

$$8 * 4 = 32$$

$$8 * 5 = 40$$

$$8 * 6 = 48$$

$$8 * 7 = 56$$

$$8 * 8 = 64$$

$$8 * 9 = 72$$

$$8 * 10 = 80$$

$$9 * 1 = 9$$

$$9 * 2 = 18$$

$$9 * 3 = 27$$

$$9 * 4 = 36$$

$$9 * 5 = 45$$

9 * 6 = 54

9 * 7 = 63

9 * 8 = 72

9 * 9 = 81

9 * 10 = 90

10 * 1 = 10

10 * 2 = 20

10 * 3 = 30

10 * 4 = 40

10 * 5 = 50

10 * 6 = 60

10 * 7 = 70

10 * 8 = 80

10 * 9 = 90

10 * 10 = 100

6.//for loop(print numbers from 1 tp 10)

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

```
int main()
```

```
{
```

```
    int i;
```

```
    for(i=1;i<=10;i++)
```

```
    {
```

```
        printf("%d \n",i);
```

```
    }
```

```
}
```



```
PS D:\learning c\output> cd 'd:\learning c\output'
PS D:\learning c\output> & .\'day6_7.exe'
1
2
3
4
5
6
7
8
9
10
PS D:\learning c\output> █
```

7. //WAP to calculate the sum of first n natural numbers

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

```
int main()
```

```
{
```

```
    int n,sum=0;
```

```
    printf("enter the limit");
```

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

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

```
    {
```

```
        sum+=i;
```

```
    }
```

```
    printf("sum is %d",sum);
```

```
}
```

```
PS D:\learning c\output> & .\'day6_8.exe'
enter the limit10
sum is 55
PS D:\learning c\output> █
```

8. /WAP to reverse a number using for loop

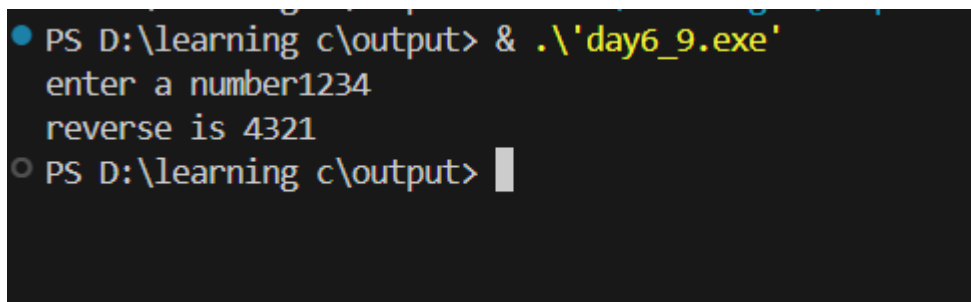
```
#include<stdio.h>
```

```
int main()
```

```

{
    int n,reverse=0;
    printf("enter a number");
    scanf("%d",&n);
    for( int i=1;n!=0;i++)
    {
        int digit=n%10;
        reverse=reverse*10+digit;
        n=n/10;
    }
    printf("reverse is %d",reverse);
}

```



```

PS D:\learning c\output> & .\'day6_9.exe\'
enter a number1234
reverse is 4321
PS D:\learning c\output>

```

9. //print fibonacci series up to n terms using for loop

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

```
int main()
```

```
{
```

```
    int n;
```

```
    printf("enter a number");
```

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

```
    int a=0;
```

```
    int b=1;
```

```
    int c;
```

```
    if(n==1)
```

```
{
```

```
    printf("%d \n",a);
}
else if(n==2)
{
    printf("%d \n",a);

    printf("%d \n",b);
}
else
{

    printf("%d \n",a);

    printf("%d \n",b);

for(int i=2;i<n;i++)
{

    c=a+b;

    printf("%d \n",c);

    a=b;
    b=c;

}
printf("\n");
```

```
}  
}
```

```
• enter a number10  
0  
1  
1  
2  
3  
5  
8  
13  
21  
34
```

10. //infinite loop using for

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

```
int main()
```

```
{
```

```
    char i=1;
```

```
    for(;;)
```

```
    {
```

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

```
        i++;
```

```
    }
```

```
}
```

11. //for with ;

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

```
int main()
```

```
{
```

```
    char i=1;
```

```
    printf("program halted");
```

```

for(;;)
{
    ;
}

}

```

12.



//pascal's triangle using nested for loop till 8 rows

```

#include<stdio.h>

int main()
{
    int x;

    int n=8;

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

        {
            printf(" ");
        }

        for(int j=0;j<=i;j++)
        {
            if(j==0 || i==0)

```

```

    {
        x=1;
    }
    else
    {
        x=(x*(i-j+1))/j;

    }
    printf("%d ",x);

}

printf("\n");
}
}

```

```

PS D:\learning c\output> & .\'day6_13.exe'
    1
    1 1
    1 2 1
    1 3 3 1
    1 4 6 4 1
    1 5 10 10 5 1
    1 6 15 20 15 6 1
    1 7 21 35 35 21 7 1
PS D:\learning c\output> 

```

13.

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



//random number guessing

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

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

```
#include <time.h>
```

```
int main()
```

```
{
```

```
    printf("This is a guessing game \n");
```

```
    printf("I have chosen a number between 0 to 20 which you must guess \n");
```

```
    int num;
```

```
    srand(time(0));
```

```
    int random=rand()%21;
```

```
    for(int i=5;i>=1;i--)
```

```
    {
```

```
        printf("you have %d tries left \n",i);
```

```
        printf("enter a guess \n");
```

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

```
        if(num==random)
```

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: 8

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!

```

{
    printf("congragulations .you guess it \n");
    break;
}
else if(num>random)
{
    printf("sorry %d is wrong .my number is less than that \n",num);

}
else if(num<random)
{
    printf("sorry %d is wrong .my number is higher than that \n",num);

}
}

```

```

PS D:\learning c\output> & .\'day6_14.exe'
This is a guessing game
i have chosen a number between 0 to 20 which you must guess
you have 5 tries left
enter a guess
3
sorry 3 is wrong .my number is higher than that
you have 4 tries left
enter a guess
10
sorry 10 is wrong .my number is less than that
you have 3 tries left
enter a guess
9
congragulations .you guess it

```



```

PS D:\learning c\output> cd 'd:\learning c\output'
PS D:\learning c\output> & .\'day6_14.exe'
This is a guessing game
i have chosen a number between 0 to 20 which you must guess
you have 5 tries left
enter a guess
1
sorry 1 is wrong .my number is higher than that
you have 4 tries left
enter a guess
3
sorry 3 is wrong .my number is higher than that
you have 3 tries left
enter a guess
1
sorry 1 is wrong .my number is higher than that
you have 2 tries left
enter a guess
3
sorry 3 is wrong .my number is higher than that
you have 1 tries left
enter a guess
3
sorry 3 is wrong .my number is higher than that
the random number is 5
PS D:\learning c\output>

```

}

14.

```

main.c
1 Problem Statement: Filter Even Numbers with Continue
2
3 Description: Write a C program that prompts the user to enter a series of integers (up to a
4 maximum of 20). The program should calculate and display the sum of all even numbers entered
5 while skipping any negative numbers. Use the continue statement to skip processing for
6 negative numbers.
7
8 Requirements:
9 1. Prompt the user for up to 20 integers.
10 2. Use a loop to read each integer.
11 3. If an integer is negative, use continue to skip adding it to the sum.
12 4. If an integer is even, add it to a running total sum.
13 5. After all inputs, display the total sum of even numbers.
14
15 Example Input/Output:
16 Enter up to 20 integers (enter -1 to stop):
17 4
18 7
19 -3
20 2
21 8
22 -5
23 10
24 -1
25 Sum of even numbers: 24

```

//filter even numbers with continue

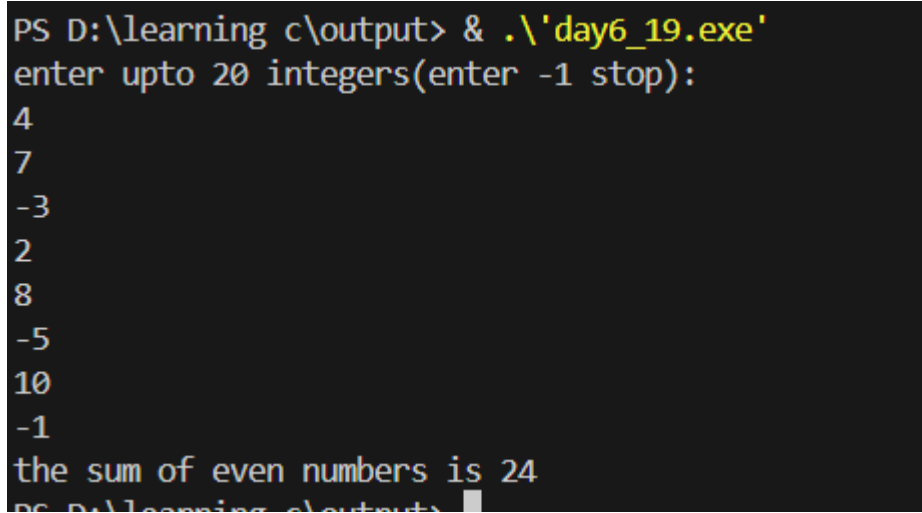
```

#include<stdio.h>

int main()
{
    int num,count=0,sum=0;;
    printf("enter upto 20 integers(enter -1 stop):");

    while(count<20)
    {
        scanf("%d",&num);
        if(num== -1)
        {
            break;
        }
        count++;
        if(num<0 || num%2!=0)
        {
            continue;
        }
        sum+=num;
    }
    printf("the sum of even numbers is %d",sum);
}

```



```

PS D:\learning c\output> & .\'day6_19.exe'
enter upto 20 integers(enter -1 stop):
4
7
-3
2
8
-5
10
-1
the sum of even numbers is 24
PS D:\learning c\output>

```

15.

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 (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.

//simple banking system

#include<stdio.h>

int main()

{

int option;

int accountnum;

char name[10];

float deposit,withdraw;

float balance=0.0;

while (1)

{

printf("welcome to the banking system \n");

printf("1.Create Account \n");

printf("2.Deposite Money \n");

printf("3.withdraw money \n");

printf("4.check balance \n");

```
printf("5.exit \n");  
printf("choose an option:");  
scanf("%d",&option);
```

```
switch (option)  
{
```

```
case 1:
```

```
    printf("enter account holder name: \n");  
    scanf(" %[^\n]%"c", name);  
    printf("enter account number");  
    scanf("%d",&accountnum);  
    printf("account created sucessfully ! account number:%d \n",accountnum);  
    break;
```

```
case 2:
```

```
    printf("enter account number \n");  
    scanf("%d",&accountnum);  
    printf("enter amount to deposit \n");  
    scanf("%f",&deposit);  
    balance+=deposit;  
    printf("deposit sucessful!! new balance:%f \n",balance);  
    break;
```

```
case 3:
```

```
    printf("enter account number \n");  
    scanf("%d",&accountnum);  
    printf("enter amount to withdraw \n");  
    scanf("%f",&withdraw);  
    if(balance<withdraw)
```

```
{  
    printf("invalid balance \n");  
    break;  
}  
balance-=withdraw;  
printf("deposit sucessful!! new balance:%f \n",balance);  
break;  
case 4:  
    printf("enter account number \n");  
    scanf("%d",&accountnum);  
    printf("current balance is %f \n",balance);  
    break;  
case 5:  
    printf("exiting the system");  
  
    return 0;  
  
default:  
    break;  
}  
}
```

```
PS D:\learning c\output> & .\'day6_20.exe'
welcome to the banking system
1.Create Account
2.Deposite Money
3.withdraw money
4.check balance
5.exit
choose an option:1
enter account holder name:
rinta
enter account number1001
account created sucessfully ! account number:1001
welcome to the banking system
1.Create Account
2.Deposite Money
3.withdraw money
4.check balance
5.exit
choose an option:2
enter account number
1001
enter amount to deposit
4000
deposit sucessful!! new balance:4000.000000
welcome to the banking system
1.Create Account
2.Deposite Money
3.withdraw money
4.check balance
5.exit
choose an option:3
enter account number
1001
enter amount to withdraw
3000
deposit sucessful!! new balance:1000.000000
```

```
welcome to the banking system
1.Create Account
2.Deposite Money
3.withdraw money
4.check balance
5.exit
choose an option:3
enter account number
1001
enter amount to withdraw
3000
deposit sucessful!! new balance:1000.000000
welcome to the banking system
1.Create Account
2.Deposite Money
3.withdraw money
4.check balance
5.exit
choose an option:4
enter account number
1001
current balance is 1000.000000
welcome to the banking system
1.Create Account
2.Deposite Money
3.withdraw money
4.check balance
5.exit
choose an option:3
enter account number
1001
enter amount to withdraw
2000
invalid balance
welcome to the banking system
1.Create Account
2.Deposite Money
3.withdraw money
4.check balance
5.exit
choose an option:5
```


choose an option:5
exiting the system

16. 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

/*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.*/

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

```
int main()
```

```
{
```

```
    float temperature[30];
```

```
float sum=0.0;

float highest,lowest;

int averagec=0;

printf("Enter temperatures for each day of the month (30 days):\n");
```

```
for (int i = 0; i < 30; i++) {

    printf("Day %d temperature: ", i + 1);

    scanf("%f", &temperature[i]);

    sum+=temperature[i];

    if(i==0)

    {

        highest=temperature[i];

        lowest=temperature[i];

    }

    if(temperature[i]>highest)

    {

        highest=temperature[i];

    }

    if(temperature[i]<lowest)

    {

        lowest=temperature[i];

    }

}
```

```
float avg=sum/30;

for (int i = 0; i < 30; i++)

{

    if(temperature[i]>averagec)

    {

        averagec++;

    }

}
```

```
}  
printf("average temperature of month :%f \n ",avg);  
printf("highest temperature :%f \n",highest);  
printf("lowest temperature:%f \n",lowest);  
printf("number of days above average temperature:%d \n",averagec);  
  
}
```

```
Day 1 temperature: 21
Day 2 temperature: 21
Day 3 temperature: 22
Day 4 temperature: 23
Day 5 temperature: 24
Day 6 temperature: 25
Day 7 temperature: 26
Day 8 temperature: 21
Day 9 temperature: 22
Day 10 temperature: 23
Day 11 temperature: 24
Day 12 temperature: 21
Day 13 temperature: 22
Day 14 temperature: 23
Day 15 temperature: 24
Day 16 temperature: 26
Day 17 temperature: 27
Day 18 temperature: 28
Day 19 temperature: 29
Day 20 temperature: 21
Day 21 temperature: 24
Day 22 temperature: 23
Day 23 temperature: 25
Day 24 temperature: 26
Day 25 temperature: 27
Day 26 temperature: 28
Day 27 temperature: 29
Day 28 temperature: 21
Day 29 temperature: 21
Day 30 temperature: 22
average temperature of month :23.966667
highest temperature :29.000000
lowest temperature:21.000000
number of days above average temperature:27
PS D:\learning c\output> █
```

17. 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.

//inventory management system

#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;
}
```

```
}
```

```
}
```

● PS D:\learning c\output> cd 'd:\learning c\output'

● PS D:\learning c\output> & .\'day6_22.exe'

1.Add Product

2.update product quantity

3.delete product

4.view all products in inventory

5.Exit

choose an option1

enter product name

basket

enter product quantity

10

enter productprice

12

1.Add Product

2.update product quantity

3.delete product

4.view all products in inventory

5.Exit

choose an option2

updating product quantity :10

1.Add Product

2.update product quantity

3.delete product

4.view all products in inventory

5.Exit

choose an option3

deleting product

1.Add Product

2.update product quantity

3.delete product

4.view all products in inventory

5.Exit

choose an option2

updating product quantity :9


```
1.Add Product
2.update product quantity
3.delete product
4.view all products in inventory
5.Exit
choose an option4
product name @
quantity : 9
price is 12.000000
1.Add Product
2.update product quantity
3.delete product
4.view all products in inventory
5.Exit
choose an option5
exiting system
PS D:\learning c\output> █
```