CSE 1112

Title: Arrays in C.

Objective:

The main objectives of this lab are to

- Print Fibonacci numbers till nth term
- Multiplication of Matrix

Theory:

An array is a variable that can store multiple values. For example, if I want to store 100 integers, I can create an array for it.

```
int data[100];
```

Declaration of an array

```
dataType arrayName[arraySize];
```

For example,

```
float mark[5];
```

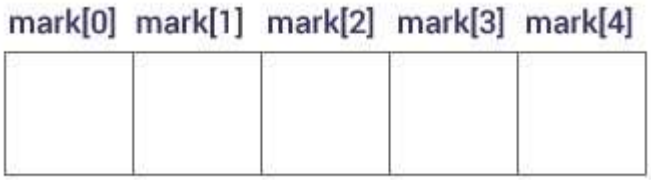
Here, I declared an array, mark, of floating-point type. And its size is 5. Meaning, it can hold 5 floating-point values.

It's important to note that the size and type of an array cannot be changed once it is declared.

Access Array Elements

I can access elements of an array by indices.

Suppose I declared an array `mark` as above. The first element is `mark[0]`, the second element is `mark[1]` and so on.



Declare an Array

Few keynotes:

- Arrays have 0 as the first index, not 1. In this example, `mark[0]` is the first element.
- If the size of an array is `n`, to access the last element, the `n-1` index is used. In this example, `mark[4]`
- Suppose the starting address of `mark[0]` is **2120d**. Then, the address of the `mark[1]` will be **2124d**. Similarly, the address of `mark[2]` will be **2128d** and so on.
  This is because the size of a `float` is 4 bytes.

Initialization of an array

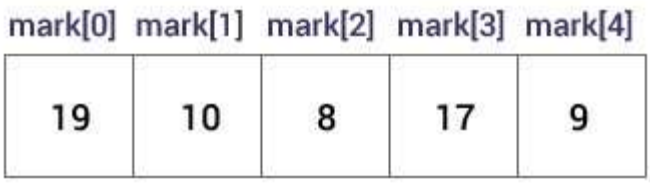
It is possible to initialize an array during declaration. For example,

```
int mark[5] = {19, 10, 8, 17, 9};
```

I can also initialize an array like this.

```
int mark[] = {19, 10, 8, 17, 9};
```

Here, I haven't specified the size. However, the compiler knows its size is 5 as we are initializing it with 5 elements.



Initialize an Array

Here,

```
mark[0] is equal to 19
mark[1] is equal to 10
mark[2] is equal to 8
mark[3] is equal to 17
mark[4] is equal to 9
```

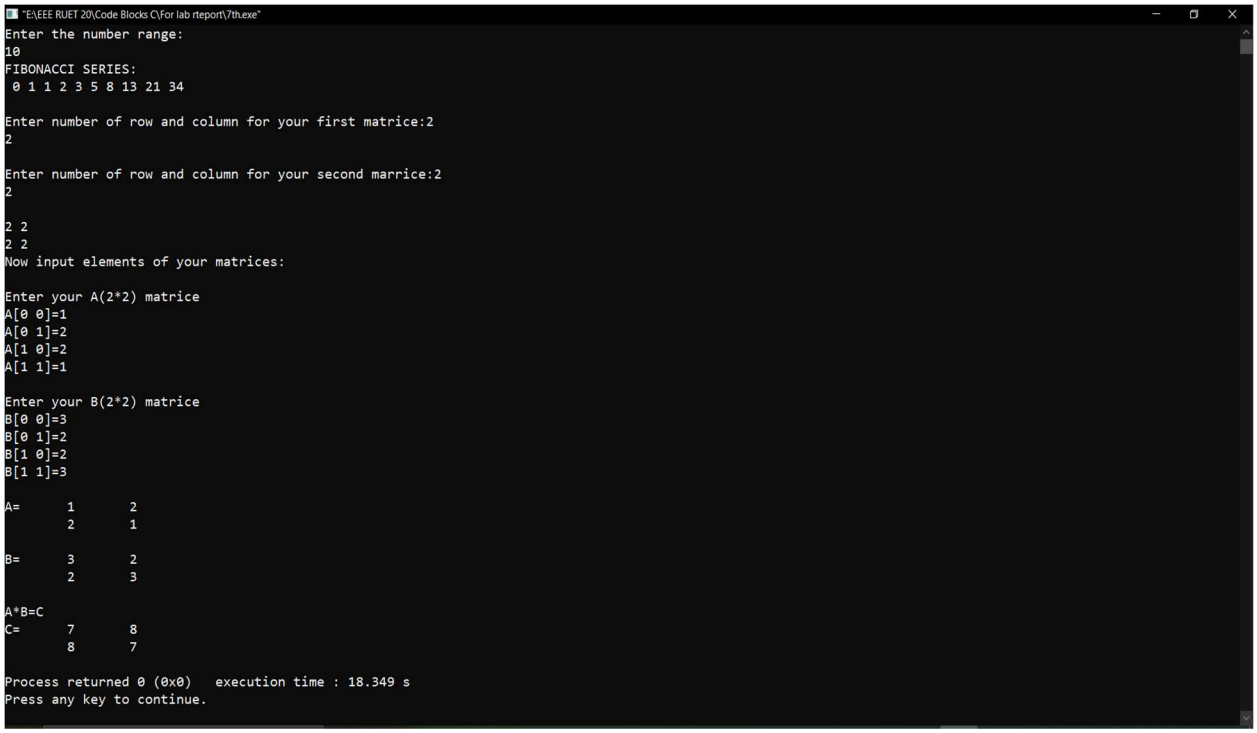
Source Code:

```
1. /// C program to print Fibonacci numbers using array.
2. #include<stdio.h>
3. int main()
4. {
5.     int n,k;
6.     long int aray[20];
7.
8.     printf("Enter the number range:\n");
9.     scanf("%d",&n);
10.         aray[0]=-1;
11.         aray[1]=1;
12.         printf("FIBONACCI SERIES: \n");
13.         for(k=2; k<=n+1; k++)
14.         {
15.             aray[k]=aray[k-1]+aray[k-2];
16.             printf(" %ld",aray[k]);
17.
18.         }
19.         printf("\n\n");
20.
21.     /// C program to display matrix multiplication using array
22.
23.         {
24.             int r1,r2,c1,c2;
25.             printf("Enter number of row and column for your
    first matrice:");
26.             scanf("%d %d",&r1,&c1);
27.             printf("\nEnter number of row and column for your
    second marrice:");
28.             scanf("%d %d",&r2,&c2);
29.             while(c1!=r2)
30.             {
31.                 printf("\nError!You cant multiply these
    matrices.Input again");
32.                 printf("\nEnter number of row and column for
    your first matrice:");
33.                 scanf("%d %d",&r1,&c1);
34.                 printf("\nEnter number of row and column for
    your second marrice:");
35.                 scanf("%d %d",&r2,&c2);
```

```
36.             }
37.             printf("\n%d %d\n%d %d",r1,c1,r2,c2);
38.             printf("\nNow input elements of your matrices:\n");
39.             int A[r1][c1],B[r2][c2],C[r1][c2];
40.             int i,j,k,sum;
41.             printf("\nEnter your A(%d*%d) matrice\n",r1,c1);
42.             for(i=0; i<r1; i++)
43.             {
44.                 for(j=0; j<c1; j++)
45.                 {
46.                     printf("A[%d %d]=",i,j);
47.                     scanf("%d",&A[i][j]);
48.                 }
49.             }
50.             printf("\nEnter your B(%d*%d) matrice\n",r2,c2);
51.             for(i=0; i<r1; i++)
52.             {
53.                 for(j=0; j<c1; j++)
54.                 {
55.                     printf("B[%d %d]=",i,j);
56.                     scanf("%d",&B[i][j]);
57.                 }
58.             }
59.             printf("\nA=");
60.             for(i=0; i<r1; i++)
61.             {
62.                 for(j=0; j<c1; j++)
63.                 {
64.                     printf("\t%d",A[i][j]);
65.                 }
66.                 printf("\n");
67.             }
68.             printf("\nB=");
69.             for(i=0; i<r2; i++)
70.             {
71.                 for(j=0; j<c2; j++)
72.                 {
73.                     printf("\t%d",B[i][j]);
74.                 }
75.                 printf("\n");
76.             }
77.             for(i=0; i<r1; i++)
78.             {
79.                 for(j=0; j<c2; j++)
80.                 {
```

```
81.                     sum=0;
82.                     for(k=0; k<r2; k++)
83.                     {
84.                         sum=sum+A[i][k]*B[k][j];
85.                         C[i][j]=sum;
86.
87.                     }
88.                 }
89.             }
90.             printf("\nA*B=C");
91.             printf("\nC=");
92.             for(i=0; i<r1; i++)
93.             {
94.                 for(j=0; j<c2; j++)
95.                 {
96.                     printf("\t%d",C[i][j]);
97.                 }
98.                 printf("\n");
99.             }
100.        }
101.        return 0;
102.    }
```

Output:



Discussion and Conclusion:

In this program, I work on Array. I write 2 programs by using array. In the first program I print Fibonacci series. It takes the term from the user then its print till that term. In the next program, it takes the sizes of matrices first then it takes input the elements of matrices from the user and finally it prints the multiplication of two matrices.