

# Arrays and Strings

Q.1: Fill in the blanks.

- (i) An \_\_\_\_\_ can store several data items of the same type.
- (ii) You refer to the individual item in an array by using an integer that is known as \_\_\_\_\_.
- (iii) A \_\_\_\_\_ array requires two separate subscript values to access an element.
- (iv) C language uses the library function \_\_\_\_\_ to read a string from the keyboard and store it in a string variable.

Answers: (i) array (ii) subscript (iii) two dimensional (iv) gets()

Q2. What is the difference between the 3s in these expression?

```
int num[3];
num[3]=5;
```

Answer:

```
int num[3]
```

Here int specified the type of elements or items that will be stored in the array and the word "num" is the name of array. The [3] following the array name tells how many variables of type int will be in our array and brackets tells the compiler that we are dealing with an array.

```
num[3]=5
```

The assignment statement stores the value 5 in the fourth location of the array because the numbering starts with 0.

Q3. Which element of the array does this expression reference?  
num [5]

Answer:

The expression reference num[5] presents 4th element of the array because the last array element will have subscript one less than the size of array.

Q.4: If an array has been defined as

```
int score[20];
```

is it correct to read values into all the elements of the array as below.

```
For(j=0; j<=20; j++)
scanf("%d", score[j]);
```

**Answer:**

There are two mistakes:

- (i) For loop counter variable  $j$  condition should be  $j \leq 19$  because the last element of array have subscript one less than the size of array.
- (ii) To read values into the elements of integer array, variable name `score` in `scanf()` statement should has an `&` (format specifier) before it to refer the address of that variable in memory.

**Q.5:** Write a program that reads  $n$  integers and prints the smallest along with its subscript value in the list.

**Answer:**

```
#include <stdio.h>
void main(void)
{
    int n, x, min, sval=0, arr[50];
    printf("Enter the number of values (Max 50)? ");
    scanf("%d", &n);

    for(x=0; x<=(n-1); x++)
    {
        printf("Enter a number: ");
        scanf("%d", &arr[x]);
    }
    min=arr[0];
    for(x=1; x<n; x++)
        if(min>arr[x])
        {
            min=arr[x];
            sval=x;
        }

    printf("The smallest number is %d", min);
    printf("\nSubscript value is %d", sval);
}
```

**OUTPUT**

```
Enter the number of values (Max 50)? 3
Enter a number: 30
Enter a number: 10
```



Enter a number: 20  
The smallest number is 10  
Subscript value is 1

Q.6: Write a program that reads two integer arrays, a and b, having 5 elements each and prints the sum of the products as given below.

$$\text{sum} = a[0]*b[0] + a[1]*b[1] + \dots + a[4]*b[4]$$

Answer:

#### PROGRAM

```
#include <stdio.h>
void main(void)
{
    int a[5], b[5], x, sum=0;
    for(x=0; x<=4; x++)
    {
        printf("Enter integer for array a[%d]:", x);
        scanf("%d", &a[x]);

        printf("Enter integer for array b[%d]: ", x);
        scanf("%d", &b[x]);
    }
    sum=a[0]*b[0]+a[1]*b[1]+a[2]*b[2]+a[3]*b[3]+a[4]*b[4];
    printf("Sum of the Products = %d", sum);
}
```

#### OUTPUT

```
Enter integer for array a[0]: 10
Enter integer for array b[0]: 10
Enter integer for array a[0]: 20
Enter integer for array b[0]: 20
Enter integer for array a[0]: 30
Enter integer for array b[0]: 30
Enter integer for array a[0]: 40
Enter integer for array b[0]: 40
Enter integer for array a[0]: 50
Enter integer for array b[0]: 50
Sum of the Products = 5500
```

Q.7: Write a program that reads n floating point numbers and prints the sum of positive numbers.

Answer:

PROGRAM

```
#include <stdio.h>
void main(void)
{
    float a,b,c,d,sum=0;
    printf("Enter the four floating point nos.");
    scanf("%f%f%f%f",&a,&b,&c,&d);
    sum=a+b+c+d;
    printf("\nthe sum is = %f", sum);
}
```

OUTPUT

Enter the four floating point nos.

10.5

20.5

30.6

34.5

the sum is = 96.099998

Q.8: For a floating-point array x whose size is n, find the geometric mean.

$$GM = \sqrt[n]{x_1 \cdot x_2 \cdot x_3 \dots x_n}$$

Answer:

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

```
#include <math.h>
```

```
void main(void)
```

```
{
```

```
    int n, v;
```

```
    float x[50];
```

```
    float gproduct=1, gmean=0;
```

```
    printf("Enter the number of values (Max 50) ? ");
```

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

 **Ratta.pk**: where students come first



```

for(v=0; v<=(n-1); v++)
{
    printf("Enter number %d: ", v+1);
    scanf("%f", &x[v]);
    gproduct=gproduct*x[v];
}
printf("\nThe product is %f", gproduct);
gmean = pow(gproduct, float(1.0/n));
printf("\nGeometric mean: %f", gmean);
}

```

## OUTPUT

Enter the number of values (Max 50) ? 3

Enter Number 1: 3

Enter Number 2: 4

Enter Number 3: 18

The product is 216.00

Geometric means: 6.00

Q.9: Write codes that will print the following patterns.

(a) 1  
1 2  
1 2 3  
1 2 3 4  
1 2 3 4 5  
(c) 1 2 3 4 5  
2 3 4 5  
3 4 5  
4 5  
5

(b) 1 2 3 4 5  
1 2 3 4  
1 2 3  
1 2  
1

(d) \*  
\* \*  
\* \* \*  
\* \* \* \*  
\* \* \* \* \*

Answer:

PROGRAM (a)

```

#include <stdio.h>
void main(void)
{
    int x, y;

```

```

for(x=1; x<=5; x++)
{
    for(y=1; y<=x; y++)
    {
        printf("%d ", y);
    }
    printf("\n");
}

```

#### PROGRAM (b)

```

#include <stdio.h>
void main(void)
{
    int x, y;
    for(x=5; x>=1; x--)
    {
        for(y=1; y<=x; y++)
        {
            printf("%d ", y);
        }
        printf("\n");
    }
}

```

#### PROGRAM (c)

```

#include <stdio.h>
void main(void)
{
    int x, y;
    for(x=1; x<=5; x++)
    {
        for(y=x; y<=5; y++)
        {
            printf("%d ", y);
        }
        printf("\n");
    }
}

```

#### PROGRAM (d)

```

#include <stdio.h>
void main(void)

```

```

{
    int x, y;
    for(x=1; x<=5; x++)
    {
        for(y=1; y<=x; y++)
        {
            printf("* ");
        }
        printf("\n");
    }
}

```

Q.10: For a two dimensional array x that has r rows and c columns, print the sum and average of each row.

Answer:

PROGRAM

```

#include <stdio.h>
void main(void)
{
    int x[2][5], r, c, sum=0, avg=0;
    for(r=0; r<=1; r++)
    {
        for(c=0; c<=4; c++)
        {
            printf("Enter value for [%d][%d] ", r, c);
            scanf("%d", &x[r][c]);
            sum=sum+x[r][c];
        }
        printf("Sum=%d, AVG=%d\n", sum, sum/5);
        sum=0;
    }
}

```

OUTPUT

```

Enter value for [0,0] 10
Enter value for [0,1] 20
Enter value for [0,2] 30
Enter value for [0,3] 40
Enter value for [0,4] 50
Sum=150, AVG=30
Enter value for [1,0] 100

```



Enter value for [1,1] 200  
Enter value for [1,2] 300  
Enter value for [1,3] 400  
Enter value for [1,4] 500  
Sum=1500, AVG=300

## Objective Questions and Answers

### Multiple Choice Questions

- Simple data types present \_\_\_\_\_ values.  
(a) Single (b) Double  
(c) Numerical (d) Dimensional
- To refer to an individual element of the array, we use \_\_\_\_\_  
(a) superscript (b) subscript  
(c) variable (d) constant
- \_\_\_\_\_ is a sequence of characters used for storing and processing words, names, addresses etc.  
(a) Array (b) Variable  
(c) String (d) Syntax
- If the marks of ten students are to be processed in a program we might use an array to \_\_\_\_\_ them.  
(a) Produce (b) Press  
(c) Store (d) Locate
- In C all the array elements are number starting at \_\_\_\_\_.  
(a) 10 (b) 0  
(c) 1 (d) 100
- The assignment statement `marks[4] = 83;` stores the value in the \_\_\_\_\_ location of the array.  
(a) 4th (b) 5th  
(c) 10th (d) 1st



7. The last array element will have subscript one less than **the** \_\_\_\_\_ of the array.
- (a) Width (b) Length  
(c) Size (d) Subscript
8. In C language, a comment begins with the \_\_\_\_\_ **slash**.
- (a) Single (b) Double  
(c) Symbol (d) None of all
9. The old fashioned comment style is useful for \_\_\_\_\_ comments.
- (a) Single line (b) Programming  
(c) Multiline (d) All a, b & c
10. The basic idea behind **exchange sort** algorithm is to continually \_\_\_\_\_ adjacent pairs of array elements.
- (a) Found (b) **Interchanged**  
(c) Good (d) **Examine**
11. What is the length of following array **"int a[] = {4,20,45,24,56}"**.
- (a) 5 (b) **4**  
(c) 6 (d) **3**
12. If the size of the **array** is **not supplied**, the compiler will \_\_\_\_\_ the number of items in **the initialization list**.
- (a) Calculate (b) **Control**  
(c) Count (d) **Comment**
13. If the number of values **supplied** are **greater** than the actual number of items then compiler will give an \_\_\_\_\_ message.
- (a) **Processed** (b) **Error**  
(c) **Positive** (d) **Numerical**
14. There are **many problems** in which the **data** being processed can be naturally organized as a \_\_\_\_\_.
- (a) Group (b) **Tool**  
(c) Table (d) **Field**

15. If the matrix has the same number of rows and columns it is called a:
- (a) Rectangular matrix
  - (b) Diagonal matrix
  - (c) Dot matrix
  - (d) Square matrix
16. { } is known as:
- (a) Bar
  - (b) Parenthesis
  - (c) Braces
  - (d) Square brackets
17. Strings are very commonly used in \_\_\_\_\_ language.
- (a) Translating
  - (b) Storing
  - (c) Processing
  - (d) Programming
18. In C language string is a data type which is an array of type \_\_\_\_\_.
- (a) character
  - (b) char
  - (c) variable
  - (d) None of all
19. Strings would enclosed in \_\_\_\_\_ quotation marks.
- (a) Double
  - (b) Single
  - (c) Tripple
  - (d) Fourth
20. "I am student" is a string \_\_\_\_\_.
- (a) Variable
  - (b) Constant
  - (c) Language
  - (d) Sequence
21. "\0" stands for a character with a numerical value of \_\_\_\_\_.
- (a) Positive
  - (b) Negative
  - (c) Zero
  - (d) None of them
22. In C language, all the strings must end with the \_\_\_\_\_ character.
- (a) Constant
  - (b) Positive
  - (c) Simple
  - (d) Null
23. Just like numeric variables C uses string variables to \_\_\_\_\_ strings.
- (a) Store
  - (b) Manipulate
  - (c) Calculate
  - (d) A & b both
24. We can use scanf() to store a string in \_\_\_\_\_ variable but it has some limitations.



- (a) numeric
- (b) string
- (c) constant
- (d) keyword

25. An \_\_\_\_\_ is simply a number of memory location each of which you can store an item of data of the same type?
- (a) Array
  - (b) One dimensional Array
  - (c) Two dimensional Array
  - (d) String
26. In one dimensional array all the data items of an array are referenced through the \_\_\_\_\_ variable name.
- (a) Same
  - (b) Different
  - (c) Common
  - (d) Multiple
27. The number of elements specified between the brackets is called the \_\_\_\_\_ of the array.
- (a) Length
  - (b) Area
  - (c) Size
  - (d) All a,b, & c
28. An array which requires a single subscript value to select an element is called.
- (a) One dimensional array
  - (b) Two dimensional array
  - (c) String
  - (d) Multidimensional array
29. Three dimensional array needs \_\_\_\_\_ subscripts values.
- (a) Two
  - (b) Three
  - (c) Four
  - (d) Many
30. An array which needs more than two subscripts values is known as:
- (a) One dimensional array
  - (b) Two dimensional array
  - (c) Three dimensional array
  - (d) Multidimensional array
31. Maximum number of elements in the array declaration `int a[5][8]` is
- (a) 28
  - (b) 32
  - (c) 35
  - (d) 40
32. If the size of the array is less than the number of initializers then,
- (a) Extra values are being ignored
  - (b) Generates an error message
  - (c) Size of Array is increased



(d) Size is neglected when values are given

33. Array subscripts in C always start at

- (a) -1 (b) 1  
(c) 0 (d) Value provided by user

34. What will happen if in a C program you assign a **value to an** array element whose subscript exceeds the size of array?

- (a) The element will be set to 0.  
(b) The compiler would report an error.  
(c) The program may crash if some important data gets overwritten.  
(d) The array size would appropriately grow.

35. What will be the output of the program?

```
#include<stdio.h>
void main(void)
{
    int a[5] = {5, 1, 15, 20, 25};
    int i, j, m;
    i = ++a[1];
    j = a[1]++;
    m = a[i++];
    printf("%d, %d, %d", i, j, m);
}
```

- (a) 2, 1, 15 (b) 1, 2, 5  
(c) 3, 2, 15 (d) 2, 3, 20

Check Your Answer

| No. | Ans. | No. | Ans. | No. | Ans. | No. | Ans. |
|-----|------|-----|------|-----|------|-----|------|
| 1.  | a    | 2.  | b    | 3.  | c    | 4.  | c    |
| 5.  | b    | 6.  | b    | 7.  | c    | 8.  | b    |
| 9.  | c    | 10. | d    | 11. | a    | 12. | c    |
| 13. | b    | 14. | c    | 15. | c    | 16. | b    |
| 17. | a    | 18. | b    | 19. | a    | 20. | b    |
| 21. | c    | 22. | d    | 23. | d    | 24. | b    |
| 25. | a    | 26. | a    | 27. | c    | 28. | a    |
| 29. | b    | 30. | d    | 31. | d    | 32. | b    |



|     |   |     |   |     |   |  |  |
|-----|---|-----|---|-----|---|--|--|
| 33. | c | 34. | c | 35. | c |  |  |
|-----|---|-----|---|-----|---|--|--|

## Short Questions and Answers

**Q.1. Define an array.**

**Ans:** A number of locations each of which can store an **item of data of the same type**. All the items of an array are referenced by the same **variable name** and each individual item is identified by a subscript in square brackets.

**Q.2. Define one dimensional array.**

**Ans:** The arrays that require a single subscript value to **select an element**, is called a one dimensional array.

**Q3. What is called the size of the array?**

**Ans:** The number of elements specified **between the brackets** is called the size of the array.

**Q.4. Which is the starting digit of all the array element in "C"?**

**Ans:** In "C" all the array elements are numbered **starting at "0"**.

**Q.5. Write down the old style C comment.**

**Ans:** The old style C comment looks like this

```
/*this is a comment*/
```

**Q.6. Why the old-fashioned comment style is useful for multi-line comments?**

**Ans:** The old-fashioned comment style is useful for multi-line comments because we don't need the comment symbol at the beginning of each line.

**Q.7. What is the formula standard deviation when a real array X whose size is n?**

**Ans:** For a real array X whose size is n, standard deviation is calculated as:

$$\text{Std -Dev} = \sqrt{\frac{\sum_{j=1}^n (x_j - \text{avg})^2}{n}}$$

Here avg is the average of n numbers.

**Q.8. What will do the compiler if the size of the array is not supplied?**

**Ans:** If the size of the array is not supplied, the compiler will count the number of items in the initialization list and fix that as the array size.

**Q.9. What will do the compiler if the number of supplied values is greater than the actual number of items?**

**Ans:** If the number of values supplied is greater than the actual number of items then the compiler will give an error message.

**Q.10. Define multidimensional array.**

**Ans:** An array that requires two or more separate subscript values to access an element, it is referred to as multidimensional array.

**Q.11. Define two dimensional array.**

**Ans:** An array that requires two subscripts to reference an element is called a two dimensional array.

**Q.12. What is three dimensional array?**

**Ans:** An array needing three subscripts is a three dimensional array.

**Q.13. Define a string.**

**Ans:** A sequence of characters is generally known as a string.

**Q.14. What is the use of strings?**

**Ans:** Strings are very commonly used in programming languages for storing and processing words names, addresses sentences, etc.

**Q.15. What is meant by subscript?**

**Ans:** It is used to refer to an individual element of an array and it is written after the array name in square brackets.