

What is the right way to initialise an array in C?

- a) `int arr{} = {1,2, 5,6,9}`
- b) `int arr[5] = {1,2, 5,6,9}`
- c) `int arr{5} = {1,2, 5,6,9}`
- d) `int arr() = {1,2, 5,6,9}`

- a) Option (a)
- b) Option (b)
- c) Option (c)
- d) Option (d)

Answer: b

An integer array of dimension 10 is declared in a C program. The memory location of the first byte of the array is 1000. What will be the location of the 9th element of the array? (Assume integer takes 4 bytes of memory and the element stored at 1000 is identified as 1st element)

- a) 1028
- b) 1032
- c) 1024
- d) 1036

- a) 1028
- b) 1032
- c) 1024
- d) 1036

Answer: b

What will be the output after execution of the program?

```
#include <stdio.h>
int main()
{
    int i, a[4]={3,1,2,4}, result;
    result=a[0];
    for(i=1; i<4; i++)
    {
        if(result<a[i])
            continue;
        result=a[i];
    }
    printf("%d", result);
    return 0;
}
```

- a) 1
- b) 2
- c) 3
- d) 4

Answer: a

Which of the statements is/are correct?

- a) An array may contain more than one element
- b) All elements of array have to be of same data type
- c) The size of array has to be declared upfront
- d) All of the above

- a) Option (a)
- b) Option (b)
- c) Option (c)
- d) Option (d)

Answer: d

What actually gets passed when you pass an array as an argument to a function

- a) Value of elements in array
- b) First element of the array
- c) Base address of the array
- d) Address of the last element of array

- a) Option (a)
- b) Option (b)
- c) Option (c)
- d) Option (d)

Answer: c

Find the output of the following C program

```
#include<stdio.h>
int main()
{
    int a;
    int arr[5] = {1, 2, 3, 4, 5};
    arr[1] = ++arr[1];
    a = arr[1]++;
    arr[1] = arr[a++];
    printf("%d, %d", a, arr[1]);
    return 0;
}
```

- a) 5, 4
- b) 5, 5
- c) 4, 4
- d) 3, 4

Answer: c

What will be the output?

```
#include <stdio.h>
int main()
{
    int p;
    int arr[10]={1,2,3,4,5,6,9,10};
    p=(arr+1)[5];
    printf("%d", p);
    return 0;
}
```

- a) 5
- b) 6
- c) 9
- d) 10

Answer: c

An array of the void data type

- a) can store any data-type
- b) only stores element of similar data type to first element
- c) acquires the data type with the highest precision in it
- d) It is not possible have an array of void data type

- a) Option (a)
- b) Option (b)
- c) Option (c)
- d) Option (d)

Answer: d

What will be the output?

```
#include<stdio.h>
int main()
{
    int n = 3;
    int sum = 4;
    switch(n)
    {
        case 2: sum = sum-2;
        case 3: sum*=5;
        break;
        default:
            sum =0;
    }
    printf("%d", sum);
    return 0;
}
```

Answer: 20

How many 'a' will be printed when the following code is executed?

```
#include <stdio.h>
int main()
{
    int i = 0;
    char c = 'a';
    while (i < 5)
    {
        i++;
        switch (c)
        {
            case 'a':
                printf("%c ", c);
                break;
        }
    }
    printf("a\n");
    return 0;
}
```

Answer: 6