

# TCS NINJA EXCEPTED TEST PAPERS-2018

The TCS test Sampus seasons will have four sections:

- 1) Section-1 **Quantitative Test**
- 2) Section-2 **Verbal Test (Test on Written English skills)**
- 3) Section-3 **Test on Programming Language Proficiency (based on 'C')**
- 4) Section-4 **Coding test (C Language)**

**1. 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.

Answer: Option C

**Explanation:**

If the index of the array size is exceeded, the program will crash. Hence "option c" is the correct answer. But the modern compilers will take care of this kind of errors.

**2. What does the following declaration mean?**

**int (\*ptr)[10];**

- A.ptr is array of pointers to 10 integers
- B.ptr is a pointer to an array of 10 integers
- C.ptr is an array of 10 integers
- D.ptr is an pointer to array

Answer: Option B

**3. In C, if you pass an array as an argument to a function, what actually gets passed?**

Jntu Test result

- 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

Answer: Option C

**Explanation:**The statement 'C' is correct. When we pass an array as a function argument, the base address of the array will be passed.

#### 4. What will be the output of the program ?

```
#include<stdio.h>

int main()

{

    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);

    return 0;

}
```

- A.2, 1, 15
- B.1, 2, 5
- C.3, 2, 15
- D.2, 3, 20

Answer: Option C

**Explanation:** Step 1: `int a[5] = {5, 1, 15, 20, 25};` The variable `arr` is declared as an integer array with a size of 5 and it is initialized to

`a[0] = 5, a[1] = 1, a[2] = 15, a[3] = 20, a[4] = 25 .`

Step 2: `int i, j, m;` The variable `i, j, m` are declared as an integer type.

Step 3: `i = ++a[1];` becomes `i = ++1;` Hence `i = 2` and `a[1] = 2`

Step 4: `j = a[1]++;` becomes `j = 2++;` Hence `j = 2` and `a[1] = 3.`

Step 5: `m = a[i++];` becomes `m = a[2];` Hence `m = 15` and `i` is incremented by 1 (`i++` means `2++` so `i=3`)

Step 6: `printf("%d, %d, %d", i, j, m);` It prints the value of the variables `i, j, m`

Hence the output of the program is 3, 2, 15

**5. Is there any difference in the following declarations?**

`int fun(int arr[]);`

`int fun(int arr[2]);`

A. Yes

B. No

Answer: Option B

**Explanation:**

No, both the statements are same. It is the prototype for the function `fun()` that accepts one integer array as a parameter and returns an integer value.

**6. Are the expressions `arr` and `&arr` same for an array of 10 integers?**

A. Yes

B. No

Answer: Option B

**Explanation:**

Both mean two different things. `arr` gives the address of the first int, whereas the `&arr` gives the address of array of ints.

**7. Which of the following statements should be used to obtain a remainder after dividing 3.14 by 2.1?**

jntu fast result

A.rem = 3.14 % 2.1;

B.rem = modf(3.14, 2.1);

C.rem = fmod(3.14, 2.1);

D.Remainder cannot be obtain in floating point division.

Answer: Option C

**Explanation:**

fmod(x,y) - Calculates x modulo y, the remainder of x/y.

This function is the same as the modulus operator. But fmod() performs floating point divisions.

**8. What are the types of pnkages?**

A.Internal and External

B.External, Internal and None

C.External and None

D.Internal

Answer: Option B

**Explanation:**External pnkage-> means global, non-static variables and functions.

Internal pnkage-> means static variables and functions with file scope.

None pnkage-> means Local variable

**9. Which of the fplowing special symbp allowed in a variable name?**

A.\* (asterisk)

B.| (pipepne)

C.-(hyphen)

D.\_\_(underscore)

Answer: Option D

**Explanation:**Variable names in C are made up of letters (upper and lower case) and digits. The underscore character ("\_") is also permitted. Names must not begin with a digit.

Examples of vapid (but not very descriptive) C variable names:

jntu fast result

=> foo

=> Bar

=> BAZ

=> foo\_bar

=> \_foo42

=> \_

=> QuUx

**10. Is there any difference between following declarations?**

1 : extern int fun();

2 : int fun();

A.Both are identical

B.No difference, except extern int fun(); is probably in another file

C.int fun(); is overridden with extern int fun();

D.None of these

Answer: Option B

**Explanation:**

extern int fun(); declaration in C is to indicate the existence of a global function and it is defined externally to the current module or in another file.

int fun(); declaration in C is to indicate the existence of a function inside the current module or in the same file.