C Pointer Basics

Question 21 CORRECT

Suppose that in a C program snippet, followings statements are used.

```
i) sizeof(int);
ii) sizeof(int*);
iii) sizeof(int**);
```

Run on IDE

Assuming size of pointer is 4 bytes and size of int is also 4 bytes, pick the most correct answer from the given options.

A Only i) would compile successfully and it would return size as 4.

i), ii) and iii) would compile successfully and size of each would be same i.e. 4

C i), ii) and iii) would compile successfully but the size of each would be different and would be decided at run time.

D ii) and iii) would result in compile error but i) would compile and result in size as 4.

C Pointer Basics **C Quiz - 101**
Discuss it

Question 21 Explanation:
Size of all pointer types is same. And whether it is a 'pointer to char' or 'pointer to int' or 'pointer to pointer to int', the size always remain same. That's why all i), ii) and iii) would compile successfully and would result in same size value of 4.

Question 22 CORRECT

Assume *int* is 4 bytes, *char* is 1 byte and *float* is 4 bytes. Also, assume that pointer size is 4 bytes (i.e. typical case)

```
char *pChar;
int *pInt;
float *pFloat;

sizeof(pChar);
sizeof(pInt);
sizeof(pFloat);
```

Run on IDE

What's the size returned for each of sizeof() operator?

A 4 4 4

B 1 4 4

C 1 4 8

D None of the above

C Pointer Basics **C Quiz - 101**
Discuss it

Question 22 Explanation:
Irrespective of the type of pointer, the size for a pointer is always same. So whether it's pointer to char or pointer to float, the size of any pointer would be same. Even size of a pointer to user defined data type (e.g. struct) is also would be same.

Question 23 WRONG

In the below statement, ptr1 and ptr2 are uninitialized pointers to int i.e. they are pointing to some random address that may or may not be valid address.

```
int* ptr1, ptr2;
```

Run on IDE

A TRUE

FALSE

C Pointer Basics **C Quiz - 108**
Discuss it

Question 23 Explanation:
Even though * is placed closer to *int*, * would be associated to ptr1 only but not to ptr2. It means that “int* ptr1” is equal to “int *ptr1”. That’s why only ptr1 is uninitialized pointer to int. Basically, though both ptr1 and ptr2 are uninitialized variables yet ptr1 is pointer to int while ptr2 is variable of type int. If we really want to make both variables as pointers, we need to mention them as “int *ptr1, *ptr2;”

Question 24 CORRECT

Pick the best statement for the following program snippet:

```
#include <stdio.h>

int main()
{
 int var;  /*Suppose address of var is 2000 */

 void *ptr = &var;
 *ptr = 5;
 printf("var=%d and *ptr=%d",var,*ptr);

 return 0;
}
```

Run on IDE

A It will print “var=5 and *ptr=2000”

B It will print “var=5 and *ptr=5”

C It will print “var=5 and *ptr=XYZ” where XYZ is some random address

Compile error

C Pointer Basics **C Quiz - 111**
Discuss it

Question 24 Explanation:

Key point in the above snippet is dereferencing of void pointer. It should be noted that dereferencing of void pointer isn't allowed because void is an incomplete data type. The correct way to assign value of 5 would be first to typecast void pointer and then use it. So instead of *ptr*, one should use *(int *)ptr*. Correct answer is d.

Question 25 CORRECT

Consider the following C program.

```
#include<stdio.h>
void mystery(int *ptra, int *ptrb)
{
   int *temp;
   temp = ptrb;
   ptrb = ptra;
   ptra = temp;
}
int main()
{
    int a=2016, b=0, c=4, d=42;
    mystery(&a, &b);
    if (a < c)
       mystery(&c, &a);
    mystery(&a, &d);
    printf("%dn", a);
}
```

Run on IDE

The output of the program _____ Note : This question was asked as Numerical Answer Type.

2016

B 0

C 4

D 8

C Pointer Basics **GATE-CS-2016 (Set 1)**
Discuss it

Question 25 Explanation:
Note that a and d are not swapped as the function mystery() doesn't change values, but pointers which are local to the function.

Question 26 CORRECT

The value printed by the following program is

```
void f(int* p, int m)
{
    m = m + 5;
    *p = *p + m;
    return;
}
void main()
{
    int i=5, j=10;
    f(&i, j);
    printf("%d", i+j);
}
```

Run on IDE

A 10

B 20

30

D 40

C Pointer Basics **GATE-CS-2016 (Set 2)**
Discuss it

Question 26 Explanation:

```
#include"stdio.h"

void f(int* p, int m)
{
    m = m + 5;
    *p = *p + m;
    return;
}
int main()
{
    int i=5, j=10;
    f(&i, j);
    printf("%d", i+j);
}
```

For i, address is passed. For j, value is passed. So in function f, p will contain address of i and m will contain value 10. Ist statement of f() will change m to 15. Then 15 will be added to value at

address p. It will make i = 5+15 = 20. j will remain 10. print statement will print 20+10 = 30. So answer is C.

Question 27 CORRECT

Consider the C program below. What does it print?

```
# include <stdio.h>
# define swapl (a, b) tmp = a; a = b; b = tmp
void swap2 ( int a, int b)
{
        int tmp;
        tmp = a; a = b; b = tmp;
 }
void swap3 (int*a, int*b)
{
        int tmp;
        tmp = *a; *a = *b; *b = tmp;
}
int main ()
{
        int num1 = 5, num2 = 4, tmp;
        if (num1 < num2) {swap1 (num1, num2);}
        if (num1 < num2) {swap2 (num1 + 1, num2);}
        if (num1 > = num2) {swap3 (&num1, &num2);}
        printf ("%d, %d", num1, num2);
}
 /* Add code here. Remove these lines if not writing code */
```

Run on IDE

A 5, 5

B 5, 4

4, 5

D 4, 4

C Pointer Basics **C Quiz - 113** **Gate IT 2008**
Discuss it

Question 27 Explanation:
"if (num1 > = num2) {swap3 (&num1, &num2);}" statement is true, so call by reference will be performed.

Question 28 WRONG

What will be the output produced by the following C code:

```
int main()
{
    int array[5][5];
    printf("%d",( (array == *array) && (*array == array[0]) ));
    return 0;
}
```

1

0

C 2

D -1

C Pointer Basics **GATE 2017 Mock**
Discuss it

Question 28 Explanation:
Given is a 2d array array[5][5].
Suppose base address of array is 2000
array = 2000
*array = 2000
array[0] = 2000
So expression is something like 2000==2000 && 2000==2000 i.e. 1&&1 will return 1.

Question 29 CORRECT

Consider the following C code

```
int main()
{
   int a = 300;
   char *b = (char *)&a;
   *++b = 2;
   printf("%d ",a);
   return 0;
}
```

Consider the size of int as two bytes and size of char as one byte. Predict the output of the following code . Assume that the machine is little-endian.

556

B 300

C Runtime Error

D Compile Time Error

C Pointer Basics **GATE 2017 Mock**
Discuss it

Question 29 Explanation:
Ans is 556 as char pointer will change the second byte of the integer in the memory and when you print it as a whole integer using %d , its value will be 556 considering little endian scenario. Please read the following link :
http://stackoverflow.com/questions/22030657/little-endian-vs-big-endian

Question 30 CORRECT

Consider the following function implemented in C:

```
void printxy(int x, int y)
{
    int *ptr;
    x = 0;
    ptr = &x;
    y = *ptr;
    *ptr = 1;
    printf("%d,%d", x, y);
}
```

The output of the printxy(1,1) is

A 0,0

B 0,1

1,0

D 1,1

C Pointer Basics GATE-CS-2017 (Set 2)
Discuss it

Question 30 Explanation:

```
#include
void main()
{
 int x = 1, y = 1;
 printxy(x,y);
}
void printxy(int x, int y)
{
    int *ptr;
    x = 0;

    // ptr point to variable of value 0
    ptr = &x;

    // y has value pointed by ptr -> x= 0;
    y = *ptr;

    // value pointed by ptr is set to 1 -> x= 1;
    *ptr = 1;

    //x be changed to 1 and y will remain 0
    printf("%d,%d", x, y);
}
```

You have completed 30/41 questions .
Your accuracy is 83%.

1 2 3 4 5

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Share this post !

23 Comments **GeeksforGeeks** 1 Login