**Understanding Pointers in C – Concepts Test**

**Roll No :** \_\_\_\_\_\_\_\_

**Marks :** \_\_\_\_\_\_\_\_

**Full Marks: 20 Time: 30 Minutes**

**Please Note:**

1. Each question has *at least* **one** correct answer, and *might* have **multiple** correct answers. Unless you can figure out ***all*** the correct answers to a particular question, ***no*** marks would be given against it.
2. To make you become confused, the questions are asked using *singular* expressions, so that you might think that there exists only one answer to a question. Therefore, be careful!
3. Assume that all programs are compiled on a **32-bit CPU**.

**1. Which of these is the reason for using pointers?**

1. To manipulate parts of arrays.
2. To refer to keywords such as for and if.
3. To return more than one value from a function.
4. To refer to particular programs more conveniently.

**2. “The address of an array is a pointer constant.”**

1. thumbup1.gifTrue
2. thumbdown.gifFalse

**3. Which is the correct way to refer to the variable** ch**, assuming the address of** ch **has been assigned to the pointer fingerch?**

1. \*fingerch
2. fingerch
3. ch
4. &ch
5. \*ch
6. \*&ch
7. \*(&ch)
8. &(\*ch)

**4. In the expression** float \*fptr; **what has type** float**?**

1. The variable fptr
2. The address of fptr
3. The variable pointed to by fptr
4. None of the above

**5. Assuming that the address of** vox **has been assigned to the pointer variable** invox**, which of the following expressions is correct?**

1. vox == &invox
2. vox == \*invox
3. invox == \*vox
4. invox == &vox

**6. Assuming that** spread[] **is a one-dimensional array of type** int**, which of the following refers to the value of the third element in the array?**

1. \*(spread + 2)
2. \*(spread + 4)
3. spread + 2
4. spread + 4

**7. What will the following program output when executed? Assume that the base address of the array is 44000.**

void main(void) {

int arr[] = {4, 5, 6};

for (int j = 0; j < 3; j++)

printf("%d ", \*(arr + j));

}

1. 4 5 6
2. 44000 44001 44002
3. 44000 44002 44004
4. 44000 44004 44008

**8. Consider the following declaration:**

char s[] = "This test is getting more and more tough... X(";

**Which of the following statements will output the text fully?**

1. printf("%s", s);
2. printf("%s", &s[0]);
3. printf("%s", &(s + 0));

**wink.gif9. When you define a string using pointer notation and the string is 10 characters long, how many bytes are set in memory? (Note: this isn’t as easy as it looks. )**

1. 10
2. 11
3. 12
4. 13

**10. “Every column of a two-dimensional array can be considered to be another two-dimensional array.”**

1. True
2. False

**11. Consider the following array definition:**

int arr[2][3] = {{10, 11, 12},

{13, 14, 15}};

**Which of the following refers to the element occupied by the number 15?**

1. arr[1][2]
2. arr[2][1]
3. \*(\*(arr + 1) + 2)
4. \*((arr + 1) + 2)
5. \*(arr[1] + 2)

**12. Consider the following program segment:**

int a = 35;

int \*b = &a;

**Which of the following statements is correct?**

1. b contains address of an int.
2. Value at address contained in b is an int.
3. b is an int pointer.
4. b points to an int.

**13. Which of the following increments the value pointed to by the pointer variable** ptr**?**

1. \*ptr++
2. (\*ptr)++
3. \*(ptr++)
4. ++\*ptr

**14. Given the declaration**

int num[] = {1, 2, 3, 4, 5};

**which of the following refers to the value 3?**

1. num[2]
2. \*(num + 2)
3. \*(2 + num)

**15. What will the following program output when executed? Assume that the base address of** a **is 22000 and the base address of** p **is 44000.**

void main(void) {

int a[] = {0, 1, 2, 3, 4};

int \*p[] = {a, a+1, a+2, a+3, a+4};

printf("%u %u %u", p, \*p, \*\*p);

}

1. 44000 0 22000
2. 22000 0 44000
3. 22000 44000 0
4. 44000 22000 0
5. 0 44000 22000
6. 0 22000 44000

**16. Which of the following statement can be described as “**func\_ptr **is a pointer to a function which returns an** int**”?**

1. int \*func\_ptr();
2. int (\*func\_ptr)();
3. int \*(func\_ptr)();

**17. Assuming we want to read in a value for** x **and the address of** x **has been assigned to** ptrx**, does the following statement look all right?**

scanf("%d", \*ptrx);

1. Well… seems to be OK…
2. Umm… I don’t think so…

**18. Suppose an array has been defined as**

int arr[3];

**Can you use the expression** arr++**?**

1. Of course!
2. Nope, never!

***For the following two questions, you have to write the answers on your own:***

**19. What statement must be added to the following program to make it work correctly?**

void main(void) {

int j, \*ptrj;

\*ptrj = 3;

}

**Answer:** ptrj = &j; **20. Read the following program carefully. The purpose of the function is to just display the array elements on the screen. The program is only partly complete. You are required to write the function** show() **on your own.**

#include <stdio.h>

**void display**(int\*); //function prototype

**void show**(int\*\*);

**main**() {

int i;

int marks[] = {55, 65, 75, 56, 78, 90};

for (i = 0; i <= 5; i++) {

display(&marks[i]);

}

}

**void display**(int \*n) {

show(&n);

}

//The function show() goes here...

void **show**(int \*\*x) {

printf("%d ", \*\*x);

}