**ItP I Quiz**

1. Given the following code

**int** A[5] = { 7, 6, 5, 4, 3 };  
**int** i = \*(A+2);

What is the value of i?

Options:

- 7  
- 6  
- **5**  
- 4

2. Given the following code:

**int** a=9, b=8, c=7;  
**int**\* A[3] = { &a, &b, &c };  
**int** res = \*\*(A+2);

What is the value of res?

Options:

- 9  
- 8  
- **7**

3. Is there a bug in the following code?

**int** f(**int** p)  
{  
 **int**\* ptr = (**int**\*)malloc(**sizeof**(**int**));  
 \*ptr = p;  
 **return** p\*2;  
}

Options:

- No bugs  
- **Yes: Memory leak: no way to access dynamic memory after returning from f**  
- Yes: Cannot allocate dynamic memory for integer values

4. How many bugs are there in the following code?

**void** f(**int** p)  
{  
 **int**\* ptr = (**int**\*)malloc(**sizeof**(**int**));  
 \*(ptr+1) = p;  
}

Options:

- 0  
- 1  
- **2** (Access to non-allocated memory, and memory leak)  
- More

5. Are there bugs in the following code?

**int**\* f(**int** p)  
{  
 **int**\* ptr = (**int**\*)malloc(2\***sizeof**(**int**));  
 \*ptr = p  
 \*(ptr+1) = p\*p;  
 **return** ptr;  
}

Options:

- **No bugs**  
- Some bugs (perhaps 1 or 2)

6. Given the following code:

**int** A[5] = { 1, 2, 3, 4, 5 };  
**int**\* ptr1 = A;  
**int**\* ptr2 = &A[4];  
**int** x = \*ptr1 + \*ptr2;

What is the value of x?

Options:

- 1  
- 8  
- **6**

7. What’s declared in the following code?

**double**\* (\*f)(**int**,**int**);

Options:

- A function with two parameters returning double  
- **A pointer to a function with two parameters returning double**

8. What entity is declared in the following code?

**struct** { **int** a, b; }\* g(**int** (\*)());

Options:

- a structure  
- a pointer to a structure  
- **a function**  
- a pointer to a function

9. Given the following code:

**int** square(**int** p) { **return** p\*p; }  
**int** sum(**int**\* p, **int** len, **int** (\*f)(**int**))  
{  
 **int** res= 0;  
 **for** (**int** i=0; i<len; i++)  
 res += f(p[i]);  
 **return** res;  
}  
**int** main()  
{  
 **int** A[5] = { 1, 2, 3, 4, 5 };  
 **int** squares = sum(A,5,square);  
}

What’s the value of squares?

Options:

- **55**  
- 78  
- 100

10. Given the following code:

**struct** S  
{  
 **int** a, b;  
};  
**struct** S s = { .a = 7, .b = 77 };  
**struct** S\* ptr1 = &s;  
**int**\* ptr2 = &s.a;  
**int** res = ptr1->b + \*ptr2;

What’s the value of res?

Options:

- 7  
- 77  
- **84**

11. Given the following code:

**struct** S  
{  
 **int** a, b;  
} s = { .a = 7, .b = 77 };

int res = s.a + (&s)->b;

What’s the value of res after the assigning?

Options:

- 7  
- 77  
- **84**

12. How many entities are declared in the following code?

**struct** S { **int** a, **int** b; } A[10];

Options:

- 1  
- 2  
- 3  
- **4**

13. Given the following declarations:

**long** square(**long** i) { **return** i\*i; }  
**typedef** **long** (\*F)(**long**);

Which declaration from the following below is **incorrect**?

- F f1 = &square;  
- F f2 = square;  
- **F f3 = square(3);**

14. Given the following code:

**long** identity(**long** i) { **return** i; }  
**long** square(**long** i) { **return** i\*i; }  
**long** cube(**long** i) { **return** i\*i\*i; }  
  
**typedef** **long** (\*F)(**long**);

F A[3] = { identity, square, cube };  
  
**long** res= 0;  
**for** (**int** i=0; i<3; i++)  
 res += A[i](i+1);

What’s the value of res after executing this code?

Options:

- 6  
- 31  
- **32**

15. What kind of entity is declared here?

**typedef** **double** (\*X)(**double**[10]);

Options:

- An array of ten doubles  
- A pointer to an array of ten doubles  
- A pointer to a function that accepts arrays of ten doubles  
- A type denoting a pointer to an array of ten doubles  
**- A type denoting a pointer to a function that accepts arrays of ten doubles**

16. What does the declaration of p mean?

**int** v;  
**int** \***const** p = &v;

Options:

- declaration of the pointer to an integer  
- **declaration of the constant pointer to an integer**  
- declaration of the pointer to a constant integer

17. What does the declaration of p mean?

**int** v;  
**const int**\* p = &v;

Options:

- declaration of the pointer to an integer  
- declaration of the constant pointer to an integer  
**- declaration of the pointer to a constant integer**

18. Find the error(s) in the following code. Reply with the number of the line with the error.

1 **double** A[5];  
2 **for** ( **int** i=0; i<5; i++)  
3 \*(A+i) = i;  
4 for ( **int** i=1; i<=5; i++ )  
5 A[i-1] = (i>1 ? A[i-2] : 1) \* A[i];

Options:

- 0 (no bugs)  
- 1 (line 2)  
- 1 (line 3)  
**-2 (lines 3 & 5)**  
- 3 (lines 3, 4, 5)

19. What is the value of x after the following code is preprocessed, compiled and executed?

**int** x, y = 7;  
#define M(a) **if** (a) x = a; **else** x = 1  
M(y>10);

Options:

- 0  
**- 1**  
- 7

20. What are differences malloc() and calloc() memory allocation functions?

Options:  
**- Both allocate memory from heap area/dynamic memory. By default,** calloc **fills the allocated memory with 0’s.**- Both allocates memory from heap area/dynamic memory. By default, malloc fills the allocated memory with 0’s.  
- The malloc() function is used to reallocate the memory to the new size, calloc() to cancel allocation.

21. Which C library should we add in order to use malloc() function?

Options:

**- stdlib.h**- stdio.h  
- strings.h

- no library is necessary

22. If a variable school is a pointer to an object of a struct type, how can we access its field named room?

Options:

school.room  
**school->room**school(room)  
school/room

23. Which kind of statements is used for specifying iterations in C?

Options:

if, switch **for, while, do**  
goto, return  
empty (null) statement

24. Which kind of statements are used for breaking the current loop iteration and to jump the next iteration?

Options:

if  
goto  
return  
break  
**continue**  
empty (null) statements

25. If school is a variable of a struct type, how can we access its field named room?

Options:

**school.room**school->room  
school(room)  
school/room

26. Question: What should be the type of *expression* in the switch statement:

**switch** ( *expression* ) *statement*

Options:

char  
float  
**int**any type

27. Which *expression* in the **for** statement is evaluated first, and only once?

**for** ( *expression1* ; *expression2* ; *expression3* ) *statement*

Options:

*expression2****expression1****expression3*all of them

28. Which expression in the **for** statement is evaluated before each of execution of the for body?

**for** ( *expression1* ; *expression2* ; *expression3* ) *statement*

Options:

***expression2****expression1  
expression3*all of them

29. Consider the following array:

**int** a[10];

Is the following expression correct for the array?

\*(2 + a) = 2;

Options:

**- correct**- incorrect  
- not sure

30. Which of the increments is executed first when used in expressions:

++a

a++

Options:

++a **increment is executed first then its value used in expressions**a++ increment is executed first then its value used in expressions  
both (a++ and ++a) increments are executed first then their value used in expressions

31. Which actions are performed by the following standard function call on the pointer ptr?

free(ptr);

Options:

- allocates memory pointed to by ptr  
- re-allocates memory pointed to by ptr  
- allocates memory pointed to by ptr, and fills memory cells with zeros  
**- releases the memory pointed to by ptr**

32. Which operator (or operators) helps us exit from a loop?

Options:

- **goto**- **return**- **break**- **all statements can be used to exit a loop**

33. What is the correct value to return to the operating system upon the successful completion of a program?

* 1
* **0**
* -1
* Programs do not return a value.

34 Which of the following is a correct comment?

* { Comment }
* **/\* Comment \*/**
* \*\* Comment \*\*
* \*/ Comment /\*

35 Which of the following is not **C** variable type?

* float
* int
* **real**
* double

36 Which of the following is the correct operator to compare two variables?

* =
* :=
* +=
* **==**
* ?=

37 Which of the following is considered TRUE in C?

* 1
* 42
* -1
* .1
* **All of the above**

38 Evaluate !(1 && !(0 || 1))

* True
* False
* Syntax error

39 Which of the loop structures guarantees to execute the body at least once?

* for
* **do … while**
* while
* all of the above
* none

40 Which declaration is **not** a proper function prototype?

* char x();
* int funct(char x, char y);
* double funct(char );
* void funct(void);
* **All of them are correct**