速成教育
SPEED UP EDUCATION

CSCA48

Introduction to Computer Science II

导师： VC

CSCA48 Quiz

SPEED UP EDUCATION

Please **read each question very carefully**, and consider your answer before marking it down on the answer sheet. **You should avoid erasing, making multiple answers, or scratching over answers** as it will cause the marking software to make mistakes on your quiz.

1.- Select which one of the options below **does not reserve lockers in the C memory model**

a) Declaring a function with a **void** return type
b) char stringy[10]=”hello!”;
c) **Declaring** a variable
d) Having input parameters in a function’s declaration
e) **Declaring** a pointer

2.- Which of the statements below is **not true** about **data types in C**

a) (int) and (double) are compatible data types and we can convert one to the other
b) (char) is compatible with (int) – we can print a char as an int and it won’t cause an error
c) We can change the data type of a specific variable by type-casting it to a different type
d) The compiler does type conversions automatically for compatible data types
e) Data types tell the compiler how to interpret binary data in the computer’s memory

3.- Which among the options below is **not** a property of boxes in the C memory model?

a) Boxes have a unique number to identify them
b) Empty boxes can have junk inside
c) Variable names are kept inside the box they are attached to
d) Boxes are reserved by the Operating System for use by our program when they are needed
e) Pointers get their own box in the memory model

4.- Which of the following data types **can not be used** as a counter in a for loop?

a) a char
b) an int
c) a double
d) a string
e) a pointer

5.- If we try to access an array like this: array[-1]=5; which of the following **can not happen.**

a) The program compiles
b) We get the last entry in the array
c) The program crashes (it is terminated by the operating system)
d) We change the value of some other variable in our program
e) The program runs and appears to work normally

6.- Which of the following steps is **not** part of the process of calling a function (including the work the function itself does)?

a) compiling the function’s code so it can run
b) reserving space and setting the values of the function’s input parameters
c) reserving space for the function’s local variables
d) updating the return value
e) releasing the memory reserved for the function

7.- Which of the following C instructions **does not reserve any boxes** in the memory model?

a) int x,y,z;
b) for (i=0; i<10; i++);
c) void func(int x);
d) double *f=NULL;
e) int main(void)

8.- Which of the options below is not something we can do with arrays? (it won’t compile!)

a) declare and initialize an array in the same line
b) for an integer array, assign an element like this: array[0]=’A’;
c) copy one array into another using ‘=’, e.g. array2=array1;
d) use an index ‘i’ that is larger than the array
e) create and use a 2-dimensional array, e.g. array[x][y]=10;

9.- If we give function_B() a pointer to a variable from function_A(), can function_B() look at other variables from function_A()?

a) yes b) no c) it’s complicated

10.- With pointers, it is possible for a function to update the value of multiple variables outside the function’s code

a) yes b) no c) it’s complicated

11.- Calling a function with the name of an array is equivalent to giving it a pointer to the first entry in the array.

a) yes b) no c) Maybe

速成教育 SPEED UP EDUCATION

Consider the following program

```
int add_one_to_letters(int array[10]) {
    int i;
    i=0;

    while (array[i]!=’\0’) {
        array[i]=array[i]+1;
        i++;
    }
    return i;
}

int main() {
    int string_length;
    char stringy[10]=”ABCDEFGHI”;

    string_length=add_one_to_letters(stringy);
    printf(“The resulting string is %s\n”,stringy);
    return 0;
}
```

12.- How many **boxes** are reserved for the program by main()?.

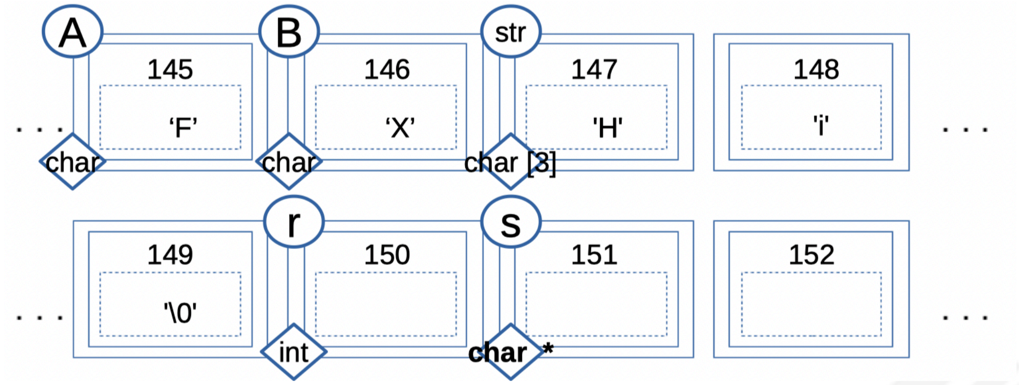
a) 4   b) 12   c) 3   d) 11   e) 13

13.- How many boxes are reserved for the program by add_one_to_letters()?

a) 4   b) 3   c) 2   d) 12   f) 11

速成教育 SPEED UP EDUCATION

Consider the memory contents shown below:



14.- What is the **value** of 's' after `s=&B;`?

a) ’X’   b) B   c) 146   d) 151   e) s

15.- Given that s=&B, what instruction would change the ‘H’ to ‘P’?

a) s+1=’P’;   b) *(s)+1=’P’;   c) *(s-1)=’P’;

d) *(s+1)=’P’;   e) &str[0]=’P’;

16.- One of the instructions below is a **bad idea** (it introduces a bug to our program). Which is it?

a) *(s)=’Y’;   b) *(s+3)=’\0’;   c) printf(“%s”,s);

d) *(s-1)=*(s+1);   e) *(s+3)=’!’;

17.- If we have s=&A; what does the instruction *(s+1)=*(s)-1; do?

a) stores a ‘Y’ in B   b) stores an ‘X’ in A   c) stores an ‘E’ in B

d) makes s=144   e) stores a 144 in B

18.- If we have s=&A; and then we have r=3; s=s+r; (instructions executed from left to right)

a) The compiler will complain   b) s will point to the box with an ‘i’   c) Program crashes

d) s will contain an invalid locker number   e) s will point to a box outside the diagram