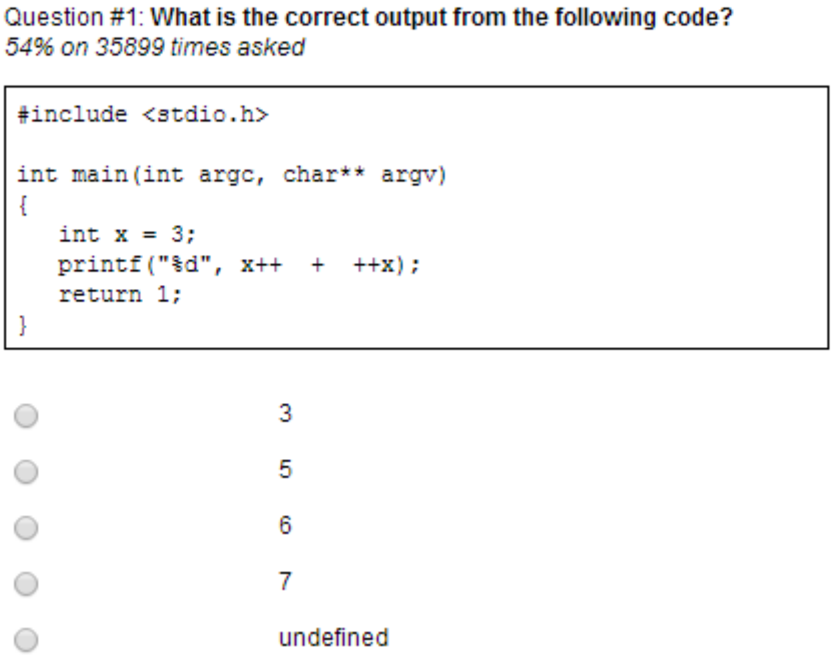
# C Quiz



Question #1: **What is the correct output from the following code?**
*54% on 35899 times asked*

```
#include <stdio.h>

int main(int argc, char** argv)
{
   int x = 3;
   printf("%d", x++  +  ++x);
   return 1;
}
```

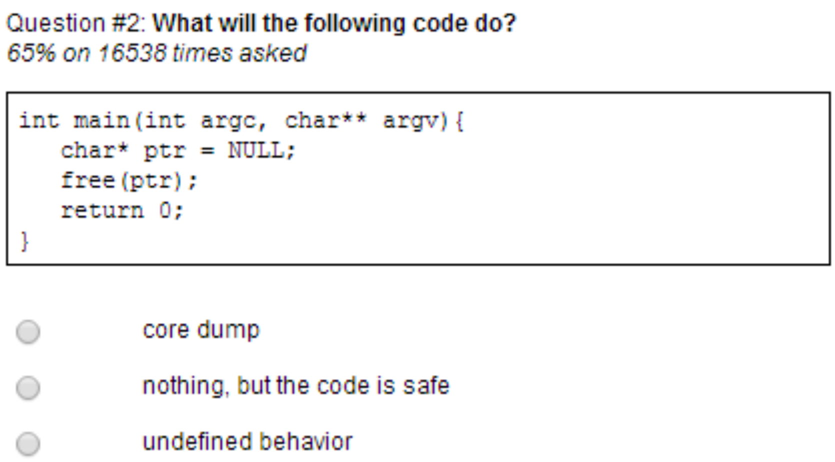
○ 3

○ 5

○ 6

○ 7

○ undefined



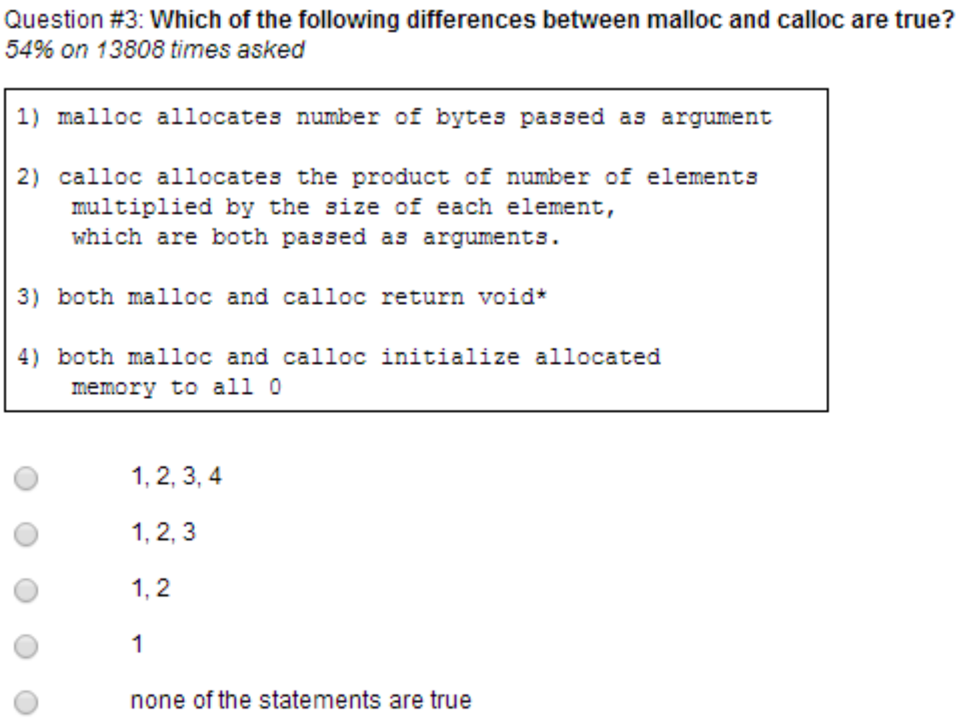
Question #2: **What will the following code do?**
*65% on 16538 times asked*

```
int main(int argc, char** argv){
   char* ptr = NULL;
   free(ptr);
   return 0;
}
```

○ core dump

○ nothing, but the code is safe

○ undefined behavior



Question #3: **Which of the following differences between malloc and calloc are true?**
*54% on 13808 times asked*

```
1) malloc allocates number of bytes passed as argument

2) calloc allocates the product of number of elements
    multiplied by the size of each element,
    which are both passed as arguments.

3) both malloc and calloc return void*

4) both malloc and calloc initialize allocated
    memory to all 0
```

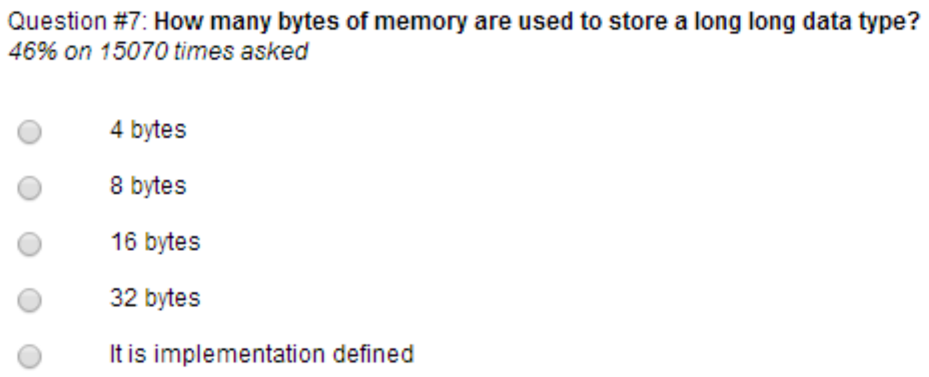
○ 1, 2, 3, 4

○ 1, 2, 3

○ 1, 2

○ 1

○ none of the statements are true



Question #7: **How many bytes of memory are used to store a long long data type?**
*46% on 15070 times asked*

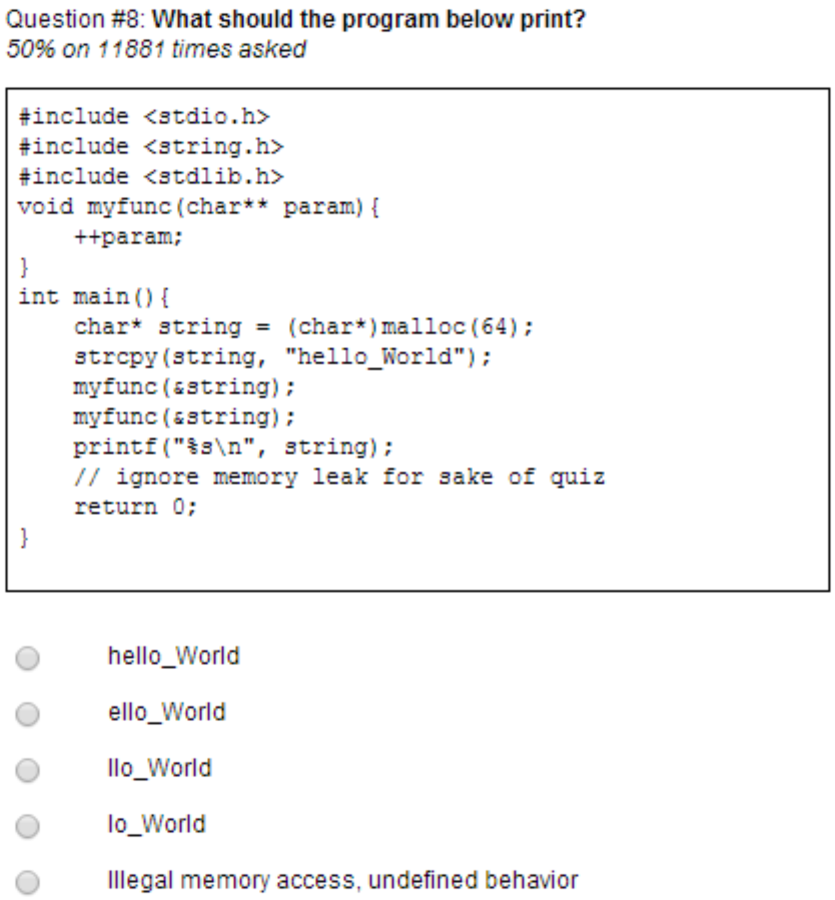
○ 4 bytes

○ 8 bytes

○ 16 bytes

○ 32 bytes

○ It is implementation defined



Question #8: **What should the program below print?**

*50% on 11881 times asked*

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
void myfunc(char** param){
    ++param;
}
int main(){
    char* string = (char*)malloc(64);
    strcpy(string, "hello_World");
    myfunc(&string);
    myfunc(&string);
    printf("%s\n", string);
    // ignore memory leak for sake of quiz
    return 0;
}
```

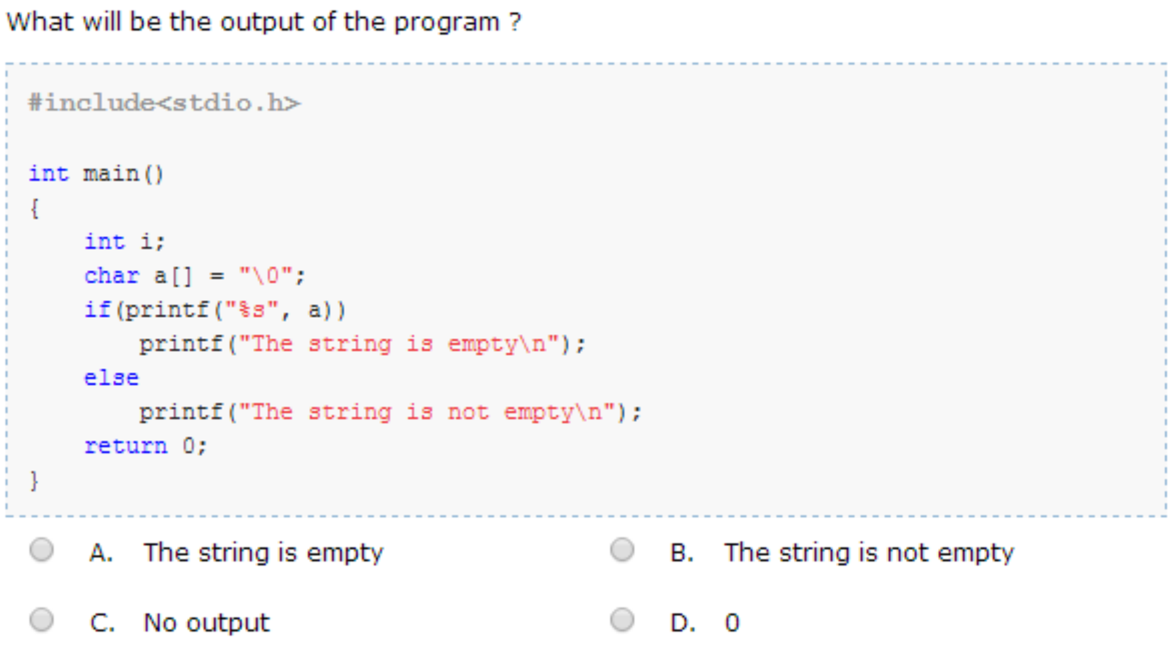
○ hello_World

○ ello_World

○ llo_World

○ lo_World

○ Illegal memory access, undefined behavior



What will be the output of the program ?

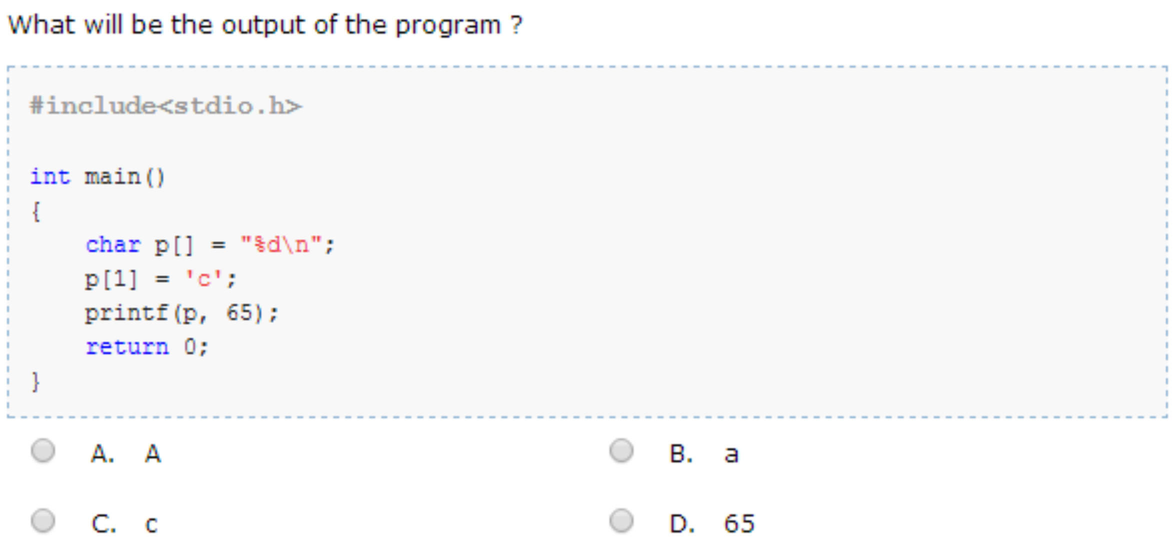
```c
#include<stdio.h>

int main()
{
    int i;
    char a[] = "\0";
    if(printf("%s", a))
        printf("The string is empty\n");
    else
        printf("The string is not empty\n");
    return 0;
}
```

○ A. The string is empty

○ B. The string is not empty

○ C. No output

○ D. 0



What will be the output of the program ?

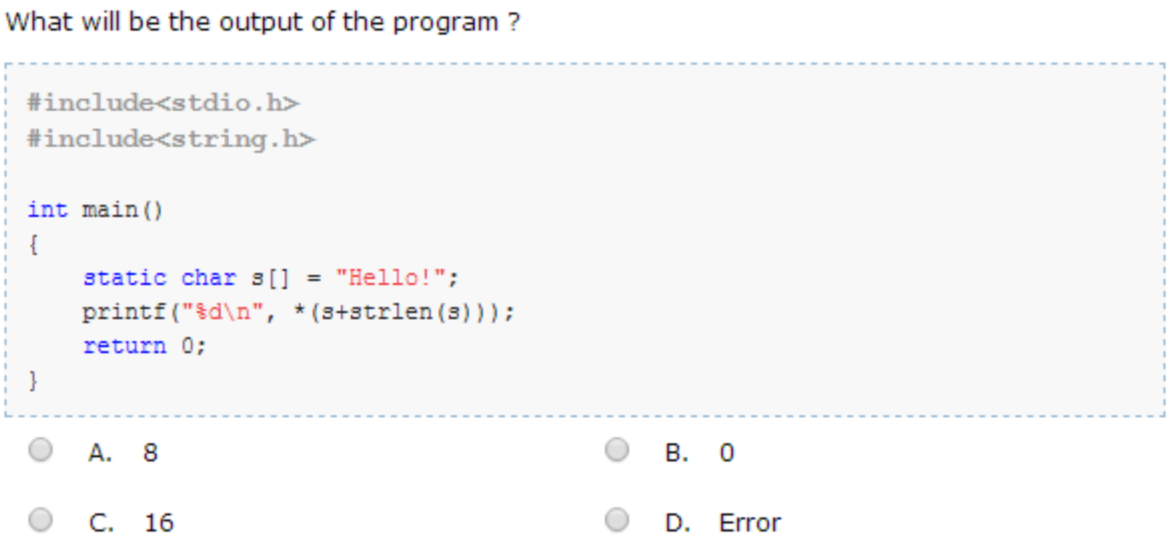
```
#include<stdio.h>

int main()
{
    char p[] = "%d\n";
    p[1] = 'c';
    printf(p, 65);
    return 0;
}
```

○ A. A

○ B. a

○ C. c

○ D. 65

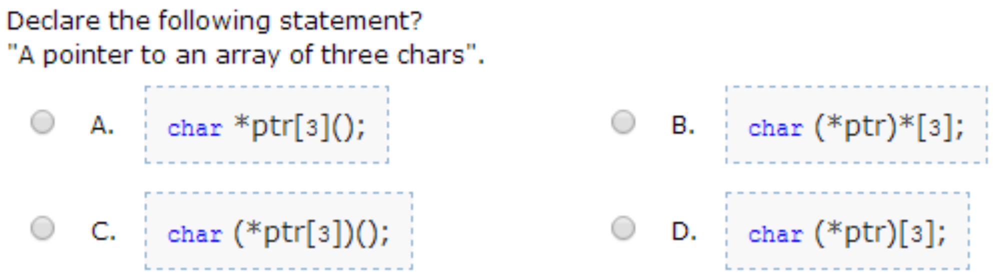


What will be the output of the program ?

```
#include<stdio.h>
#include<string.h>

int main()
{
    static char s[] = "Hello!";
    printf("%d\n", *(s+strlen(s)));
    return 0;
}
```

○ A. 8 ○ B. 0

○ C. 16 ○ D. Error



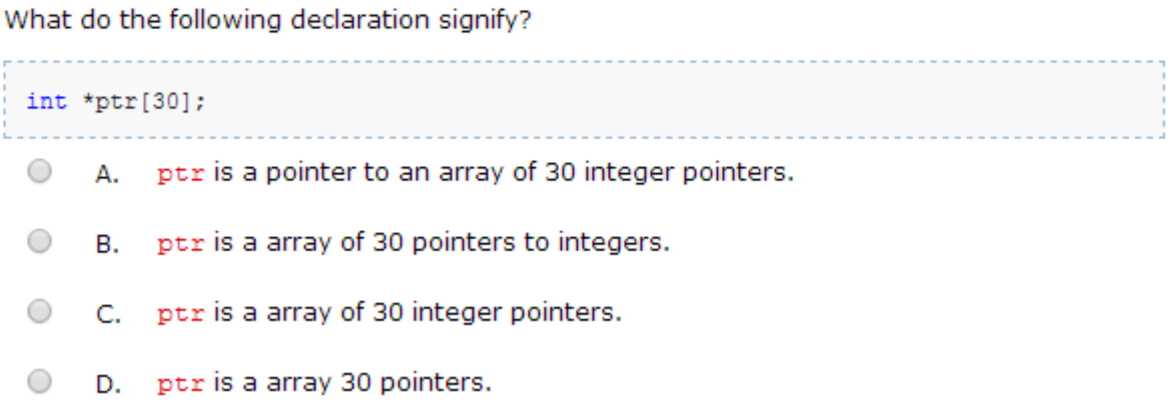
Declare the following statement?
"A pointer to an array of three chars".

○ A. `char *ptr[3]();`

○ B. `char (*ptr)*[3];`

○ C. `char (*ptr[3])();`

○ D. `char (*ptr)[3];`



What do the following declaration signify?

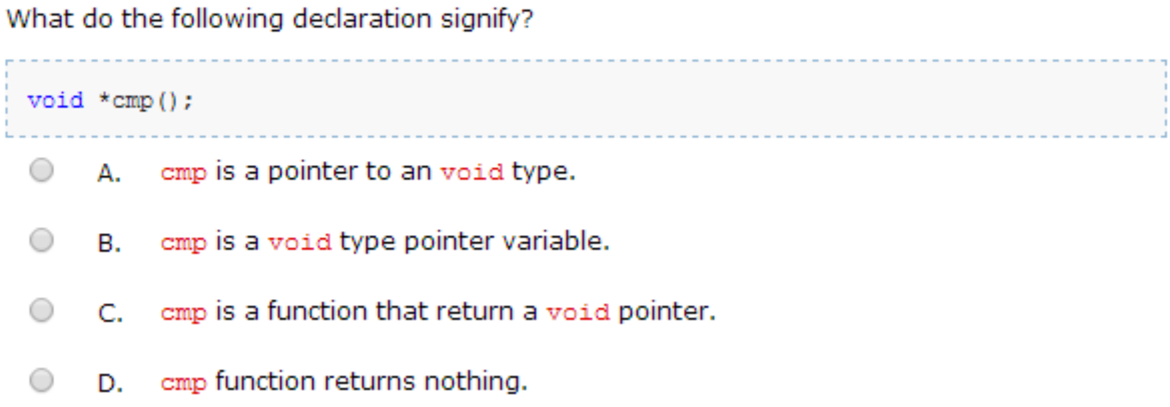
```
int *ptr[30];
```

○ A. ptr is a pointer to an array of 30 integer pointers.

○ B. ptr is a array of 30 pointers to integers.

○ C. ptr is a array of 30 integer pointers.

○ D. ptr is a array 30 pointers.



What do the following declaration signify?

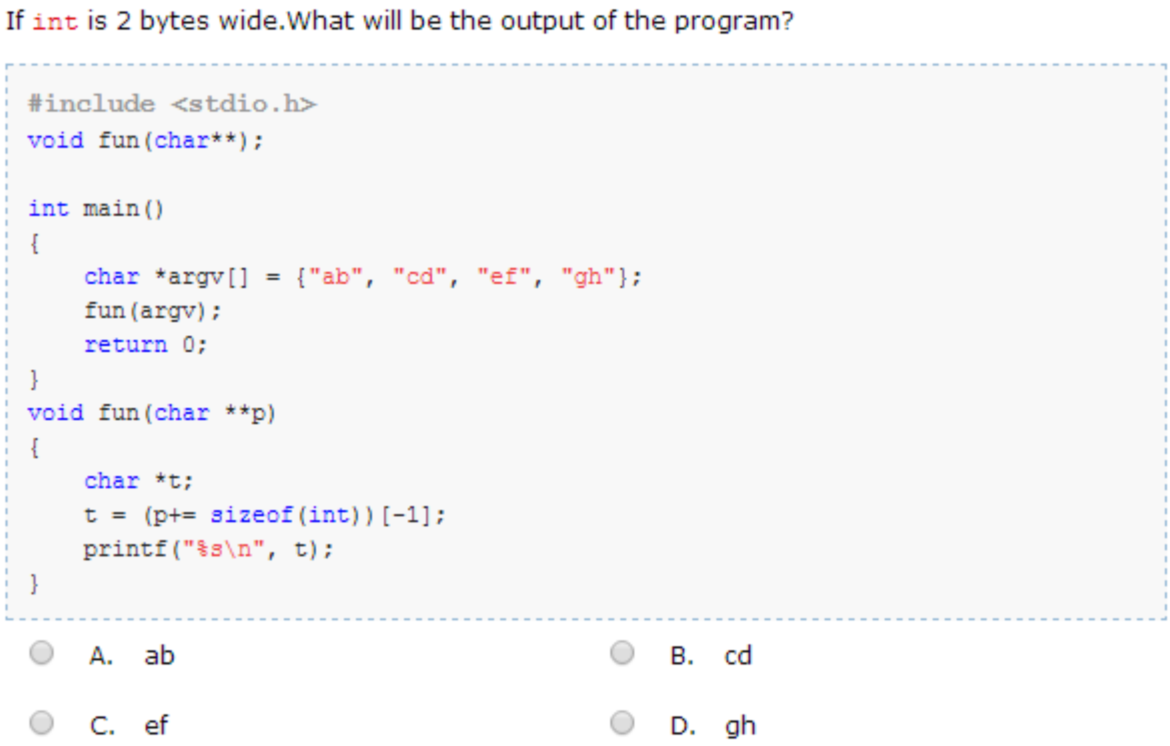
```
void *cmp();
```

○ A. cmp is a pointer to an void type.

○ B. cmp is a void type pointer variable.

○ C. cmp is a function that return a void pointer.

○ D. cmp function returns nothing.



If int is 2 bytes wide.What will be the output of the program?

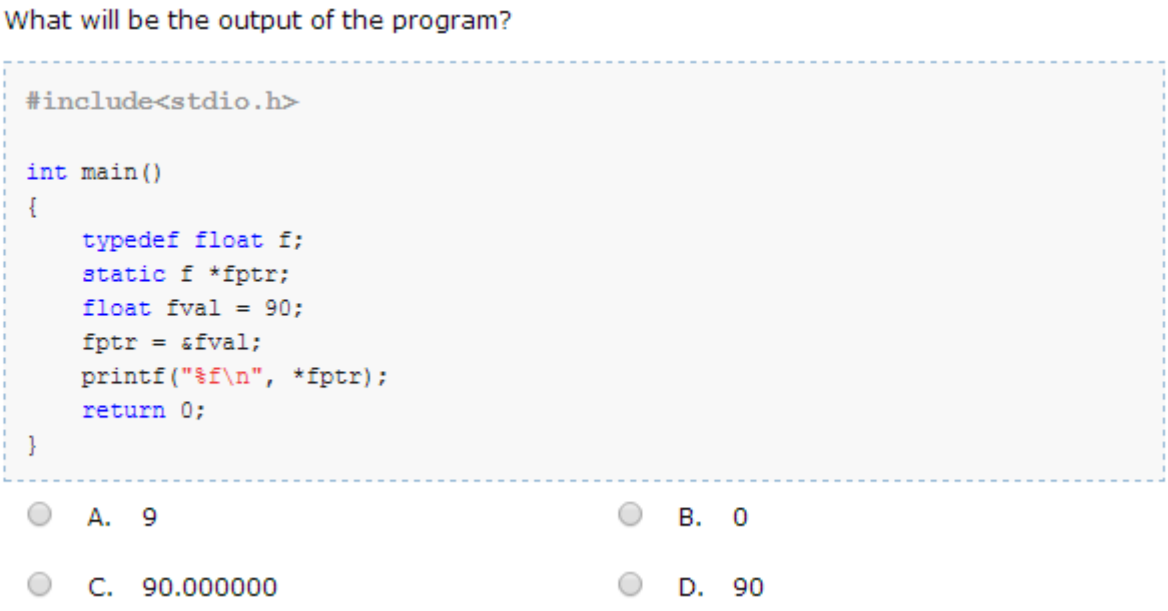
```
#include <stdio.h>
void fun(char**);

int main()
{
    char *argv[] = {"ab", "cd", "ef", "gh"};
    fun(argv);
    return 0;
}
void fun(char **p)
{
    char *t;
    t = (p+= sizeof(int))[-1];
    printf("%s\n", t);
}
```

○ A. ab

○ B. cd

○ C. ef

○ D. gh



What will be the output of the program?

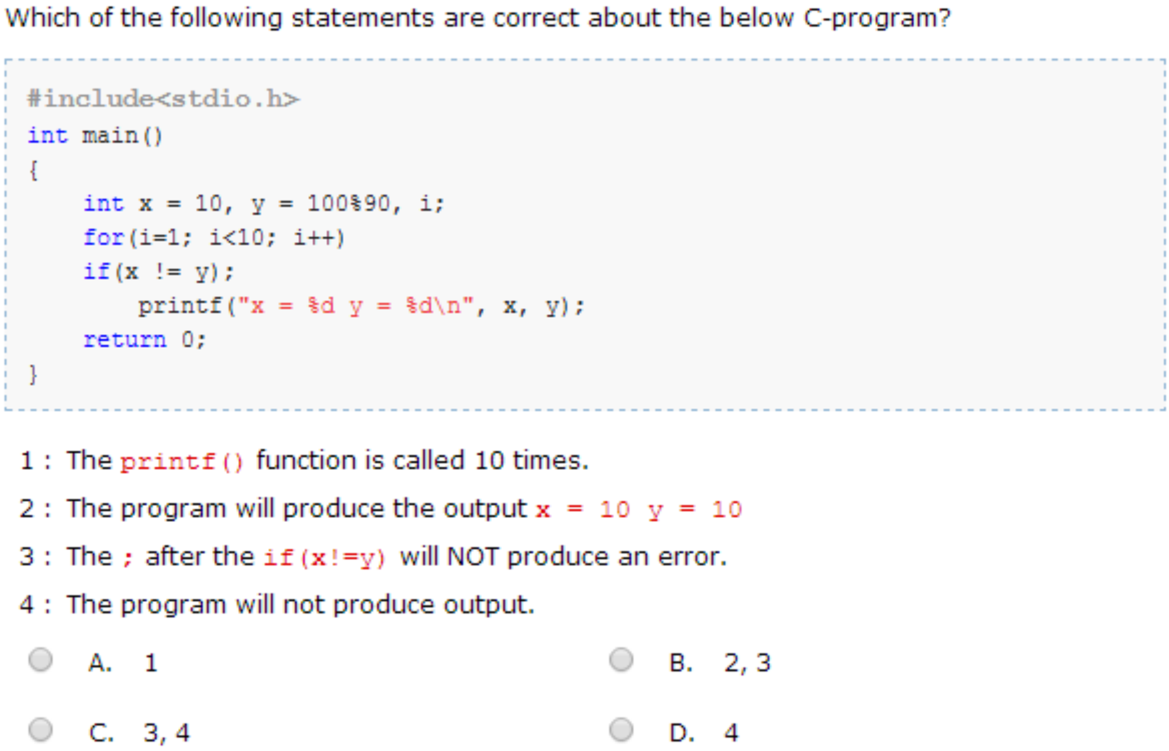
```
#include<stdio.h>

int main()
{
    typedef float f;
    static f *fptr;
    float fval = 90;
    fptr = &fval;
    printf("%f\n", *fptr);
    return 0;
}
```

○ A. 9

○ B. 0

○ C. 90.000000

○ D. 90



Which of the following statements are correct about the below C-program?

```
#include<stdio.h>
int main()
{
    int x = 10, y = 100%90, i;
    for(i=1; i<10; i++)
    if(x != y);
        printf("x = %d y = %d\n", x, y);
    return 0;
}
```

1 : The `printf()` function is called 10 times.

2 : The program will produce the output `x = 10 y = 10`

3 : The `;` after the `if(x!=y)` will NOT produce an error.

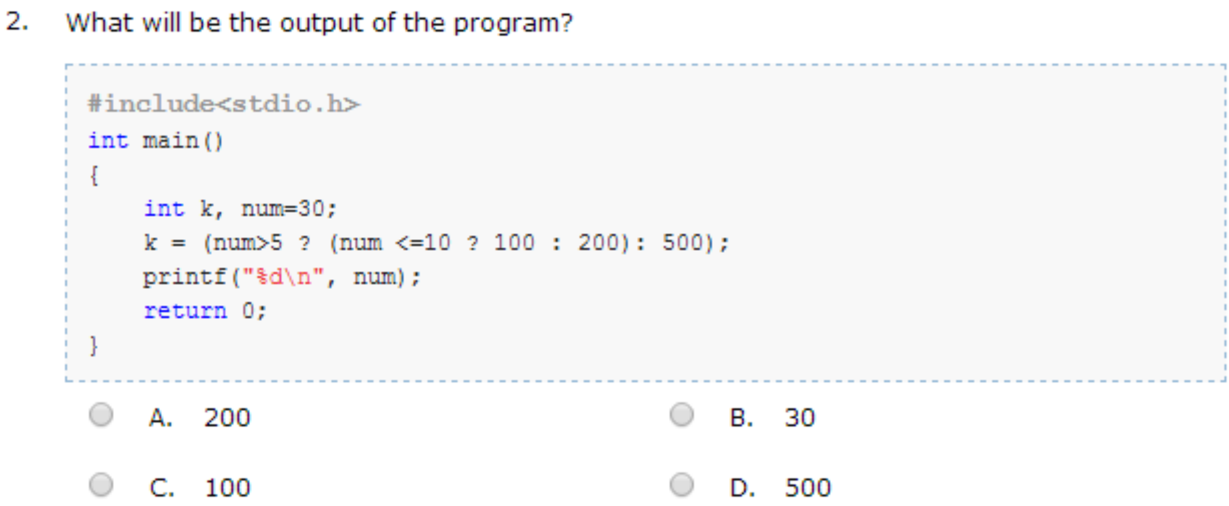
4 : The program will not produce output.

○ A. 1

○ B. 2, 3

○ C. 3, 4

○ D. 4

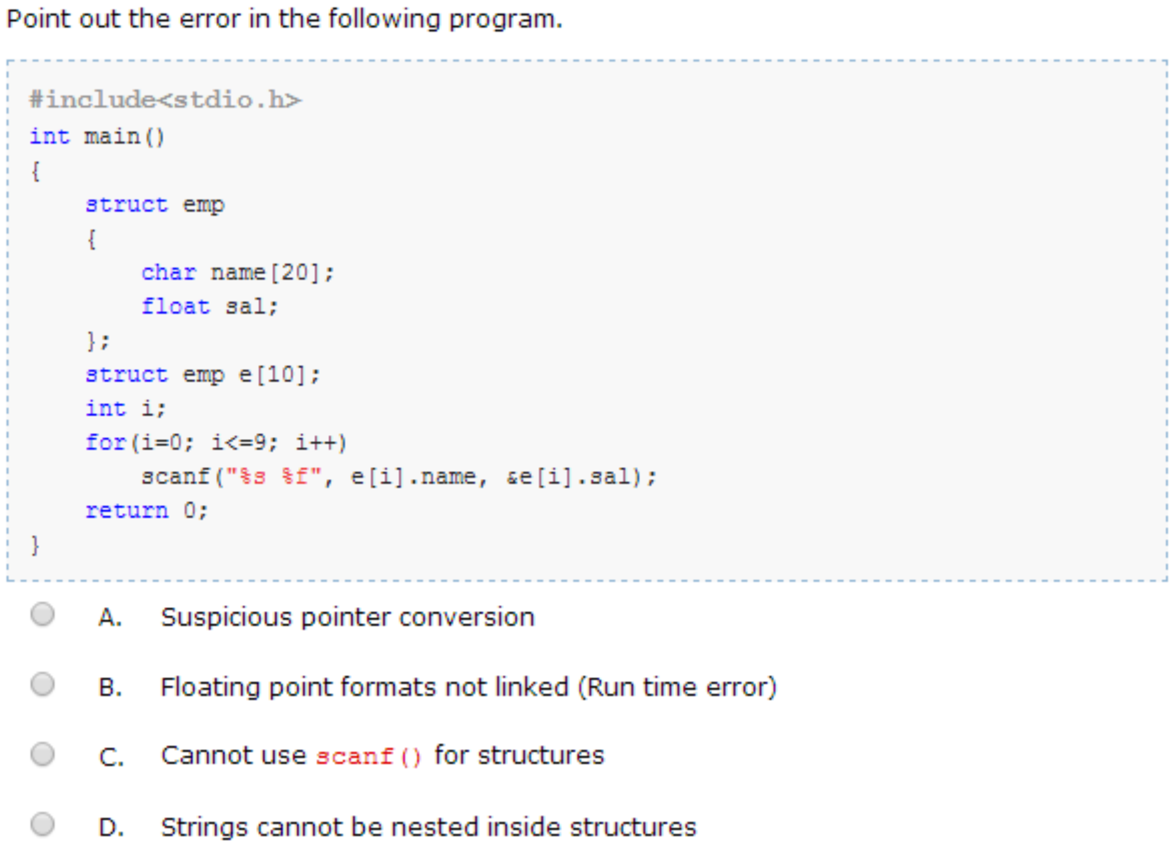


2. What will be the output of the program?

```
#include<stdio.h>
int main()
{
    int k, num=30;
    k = (num>5 ? (num <=10 ? 100 : 200): 500);
    printf("%d\n", num);
    return 0;
}
```

○ A. 200 ○ B. 30

○ C. 100 ○ D. 500



Point out the error in the following program.

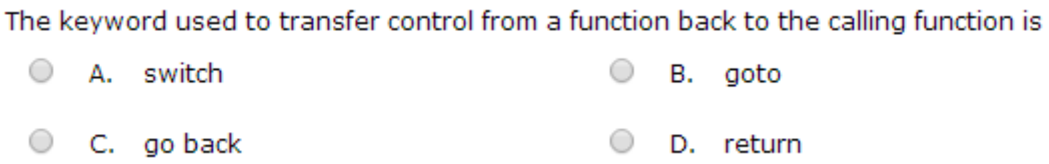
```
#include<stdio.h>
int main()
{
    struct emp
    {
        char name[20];
        float sal;
    };
    struct emp e[10];
    int i;
    for(i=0; i<=9; i++)
        scanf("%s %f", e[i].name, &e[i].sal);
    return 0;
}
```

○ A. Suspicious pointer conversion

○ B. Floating point formats not linked (Run time error)

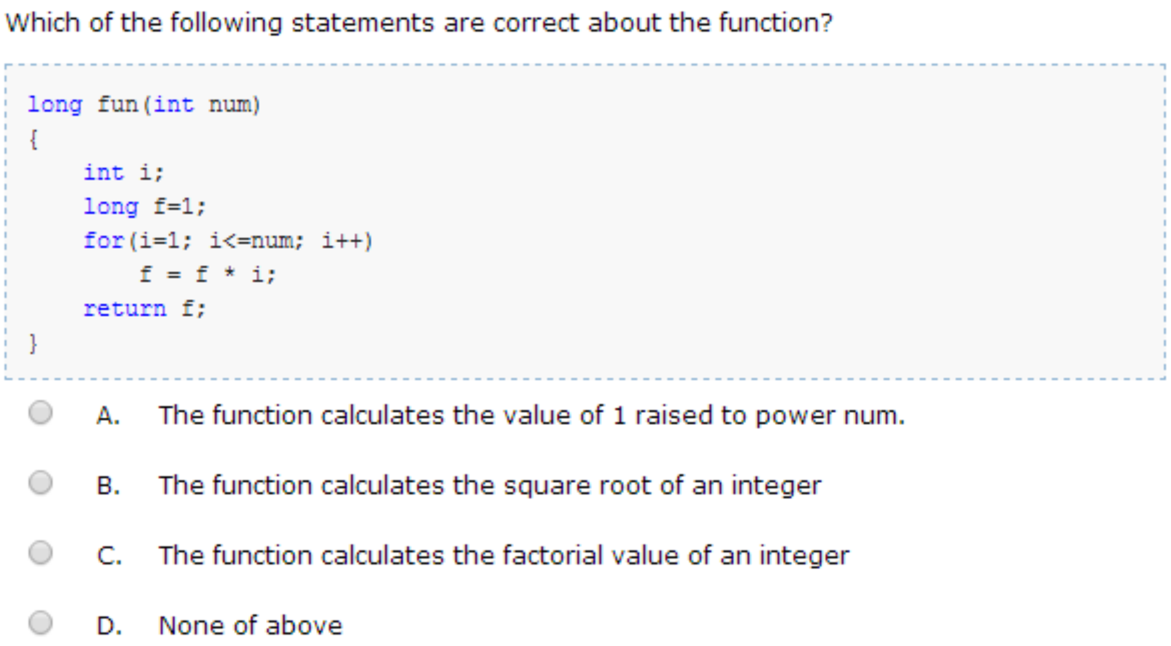
○ C. Cannot use `scanf()` for structures

○ D. Strings cannot be nested inside structures



The keyword used to transfer control from a function back to the calling function is

| | |
|---|---|
| ○ A. switch | ○ B. goto |
| ○ C. go back | ○ D. return |



Which of the following statements are correct about the function?

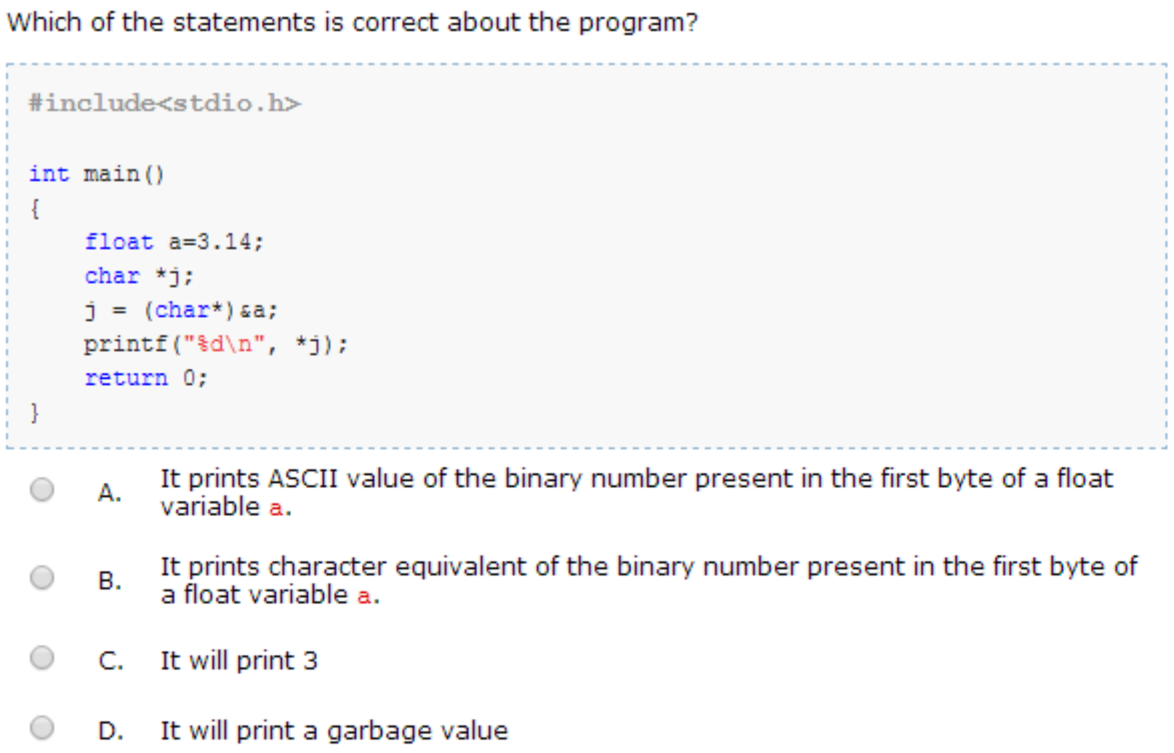
```
long fun(int num)
{
    int i;
    long f=1;
    for(i=1; i<=num; i++)
        f = f * i;
    return f;
}
```

○ A. The function calculates the value of 1 raised to power num.

○ B. The function calculates the square root of an integer

○ C. The function calculates the factorial value of an integer

○ D. None of above



Which of the statements is correct about the program?

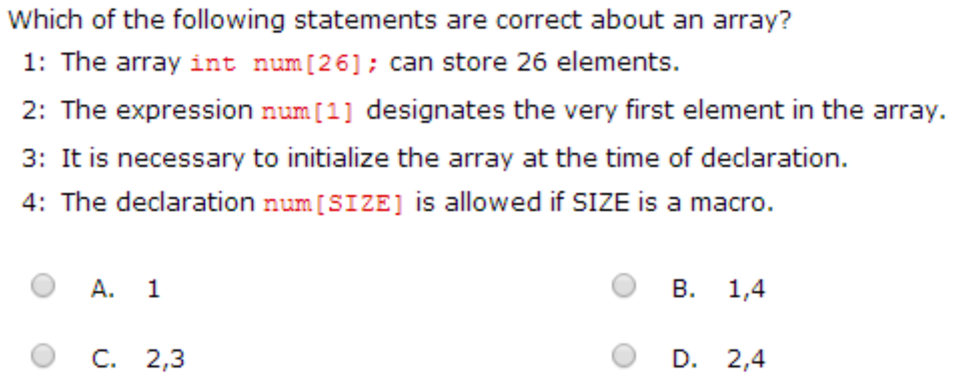
```
#include<stdio.h>

int main()
{
    float a=3.14;
    char *j;
    j = (char*)&a;
    printf("%d\n", *j);
    return 0;
}
```

○ A. It prints ASCII value of the binary number present in the first byte of a float variable a.

○ B. It prints character equivalent of the binary number present in the first byte of a float variable a.

○ C. It will print 3

○ D. It will print a garbage value



Which of the following statements are correct about an array?

1: The array `int num[26];` can store 26 elements.

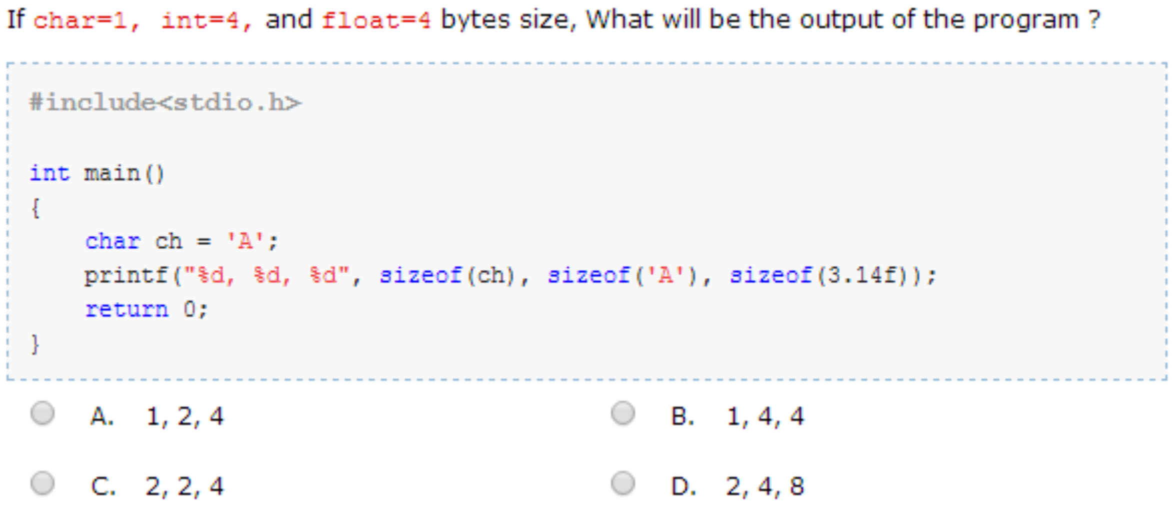
2: The expression `num[1]` designates the very first element in the array.

3: It is necessary to initialize the array at the time of declaration.

4: The declaration `num[SIZE]` is allowed if SIZE is a macro.

○ A. 1 ○ B. 1,4

○ C. 2,3 ○ D. 2,4

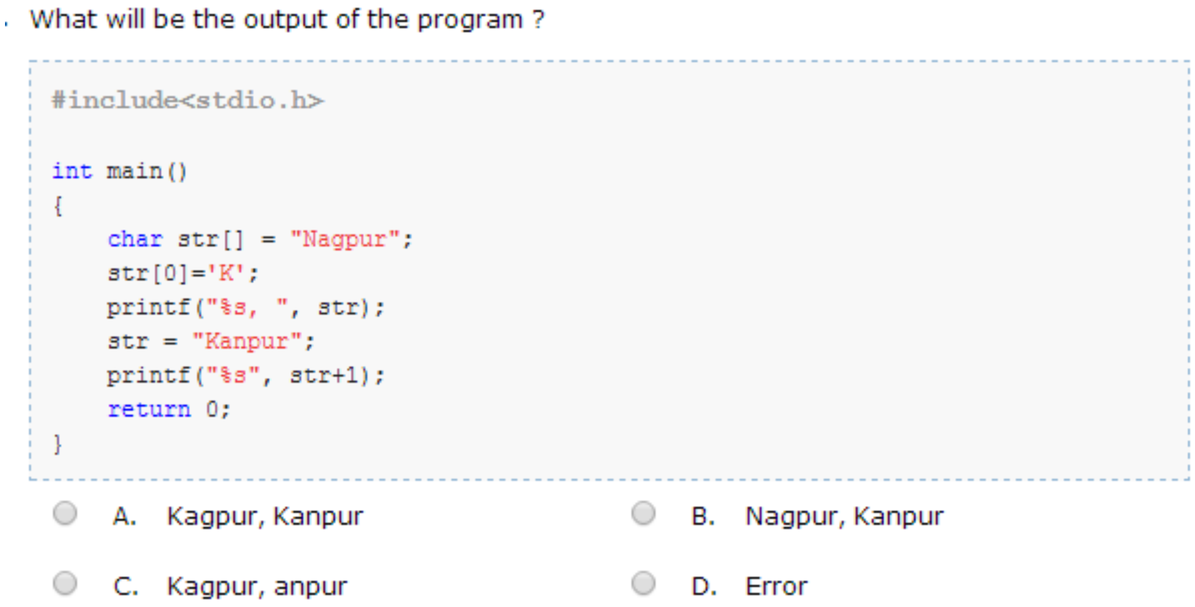


If char=1, int=4, and float=4 bytes size, What will be the output of the program ?

```
#include<stdio.h>

int main()
{
    char ch = 'A';
    printf("%d, %d, %d", sizeof(ch), sizeof('A'), sizeof(3.14f));
    return 0;
}
```

○ A. 1, 2, 4 ○ B. 1, 4, 4

○ C. 2, 2, 4 ○ D. 2, 4, 8



What will be the output of the program ?

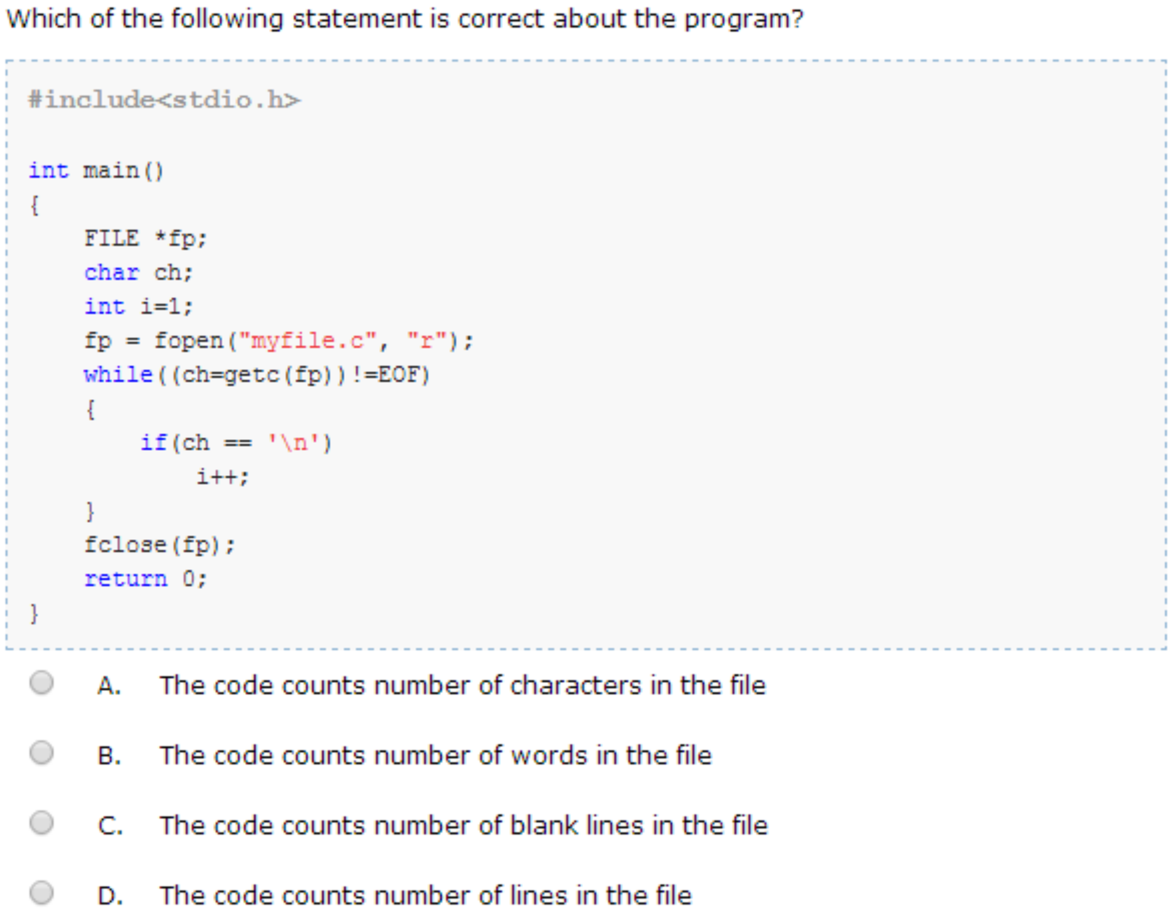
```
#include<stdio.h>

int main()
{
    char str[] = "Nagpur";
    str[0]='K';
    printf("%s, ", str);
    str = "Kanpur";
    printf("%s", str+1);
    return 0;
}
```

○ A. Kagpur, Kanpur

○ B. Nagpur, Kanpur

○ C. Kagpur, anpur

○ D. Error



Which of the following statement is correct about the program?

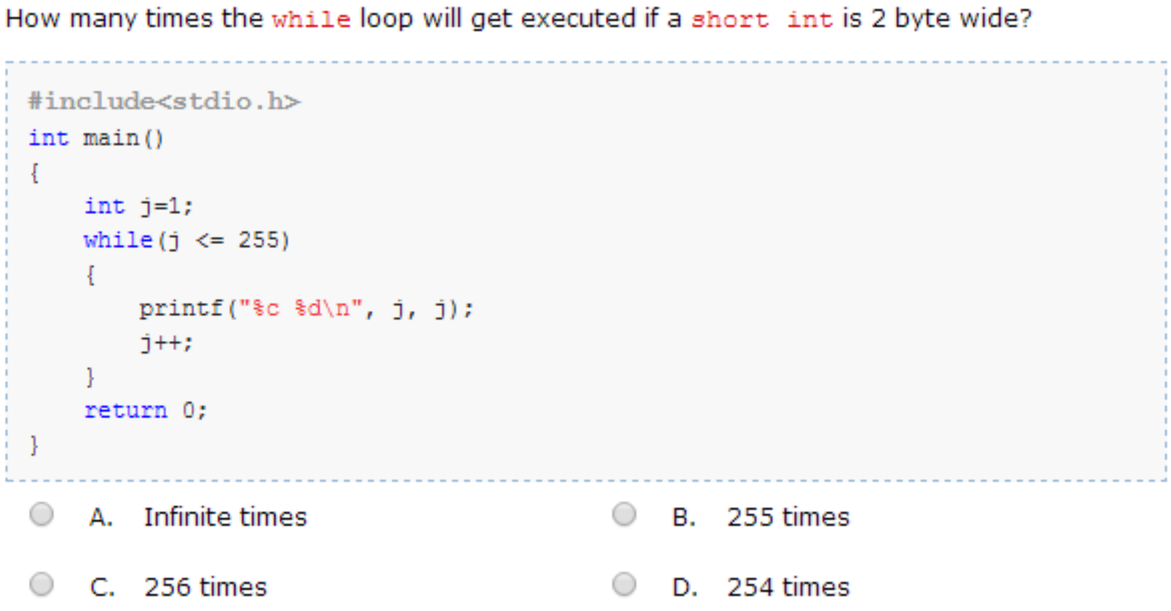
```
#include<stdio.h>

int main()
{
    FILE *fp;
    char ch;
    int i=1;
    fp = fopen("myfile.c", "r");
    while((ch=getc(fp))!=EOF)
    {
        if(ch == '\n')
            i++;
    }
    fclose(fp);
    return 0;
}
```

○ A. The code counts number of characters in the file

○ B. The code counts number of words in the file

○ C. The code counts number of blank lines in the file

○ D. The code counts number of lines in the file



How many times the `while` loop will get executed if a `short int` is 2 byte wide?

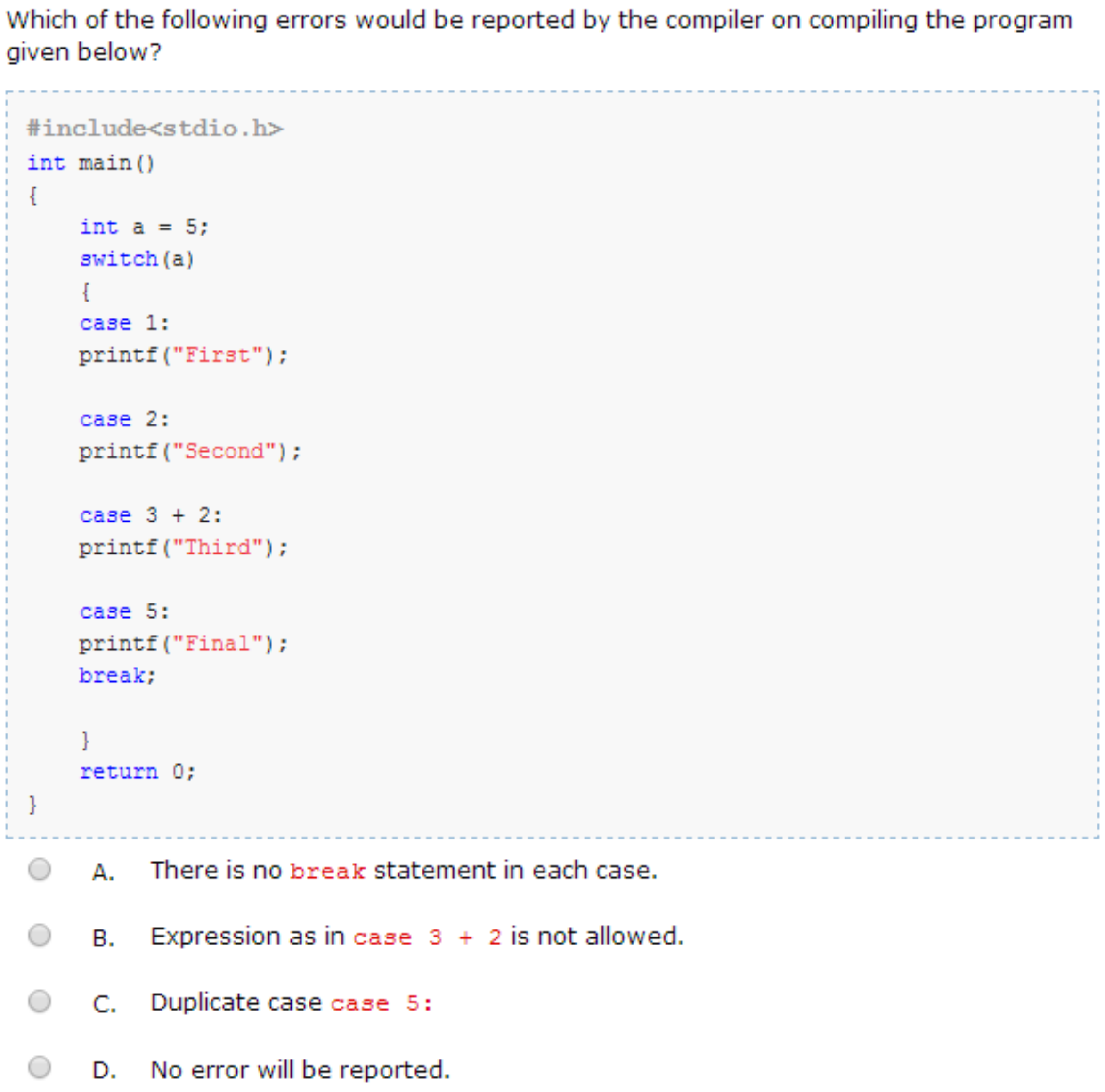
```
#include<stdio.h>
int main()
{
    int j=1;
    while(j <= 255)
    {
        printf("%c %d\n", j, j);
        j++;
    }
    return 0;
}
```

○ A. Infinite times

○ B. 255 times

○ C. 256 times

○ D. 254 times



Which of the following errors would be reported by the compiler on compiling the program given below?

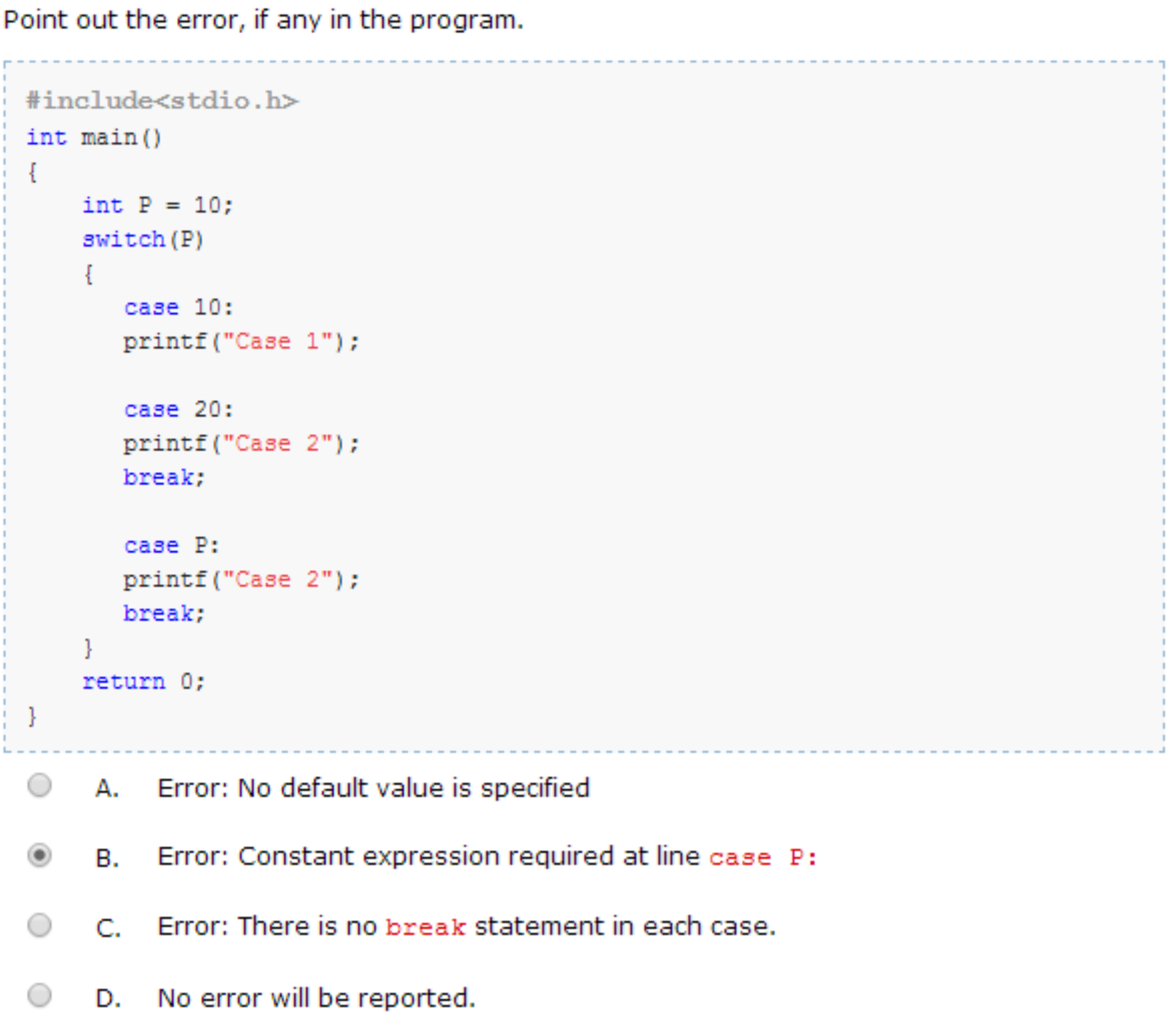
```
#include<stdio.h>
int main()
{
    int a = 5;
    switch(a)
    {
    case 1:
    printf("First");

    case 2:
    printf("Second");

    case 3 + 2:
    printf("Third");

    case 5:
    printf("Final");
    break;

    }
    return 0;
}
```

○ A. There is no `break` statement in each case.

○ B. Expression as in `case 3 + 2` is not allowed.

○ C. Duplicate case `case 5:`

○ D. No error will be reported.



Point out the error, if any in the program.

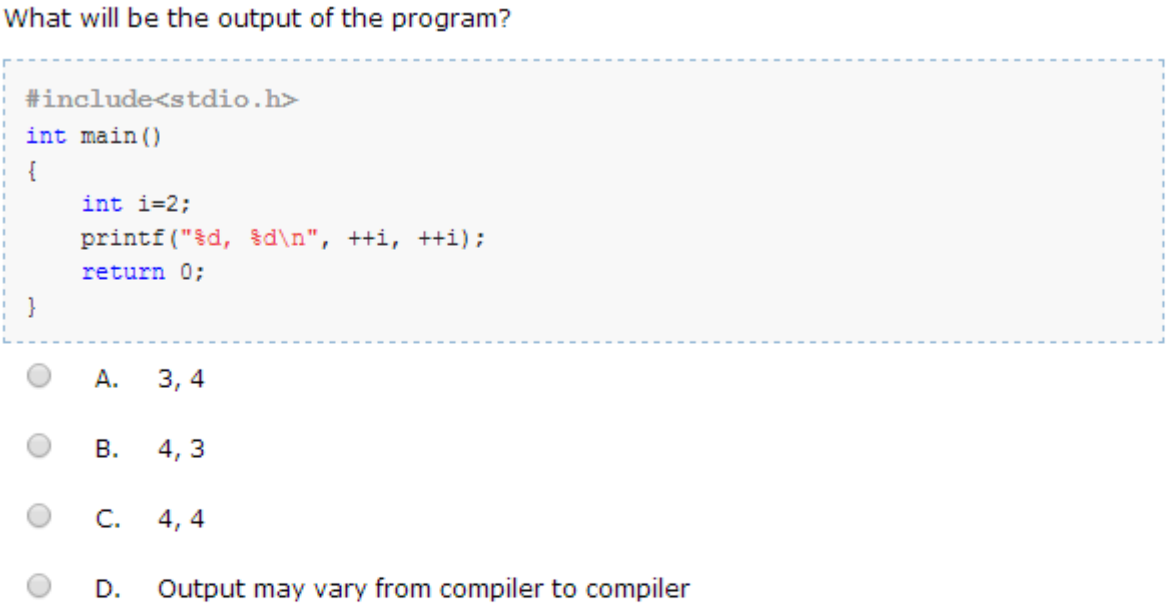
```
#include<stdio.h>
int main()
{
    int P = 10;
    switch(P)
    {
       case 10:
       printf("Case 1");

       case 20:
       printf("Case 2");
       break;

       case P:
       printf("Case 2");
       break;
    }
    return 0;
}
```

○ A. Error: No default value is specified

● B. Error: Constant expression required at line `case P:`

○ C. Error: There is no `break` statement in each case.

○ D. No error will be reported.



What will be the output of the program?

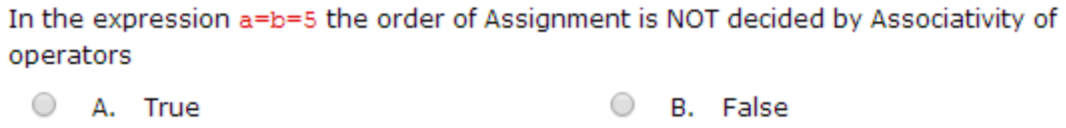
```
#include<stdio.h>
int main()
{
    int i=2;
    printf("%d, %d\n", ++i, ++i);
    return 0;
}
```

○ A. 3, 4

○ B. 4, 3

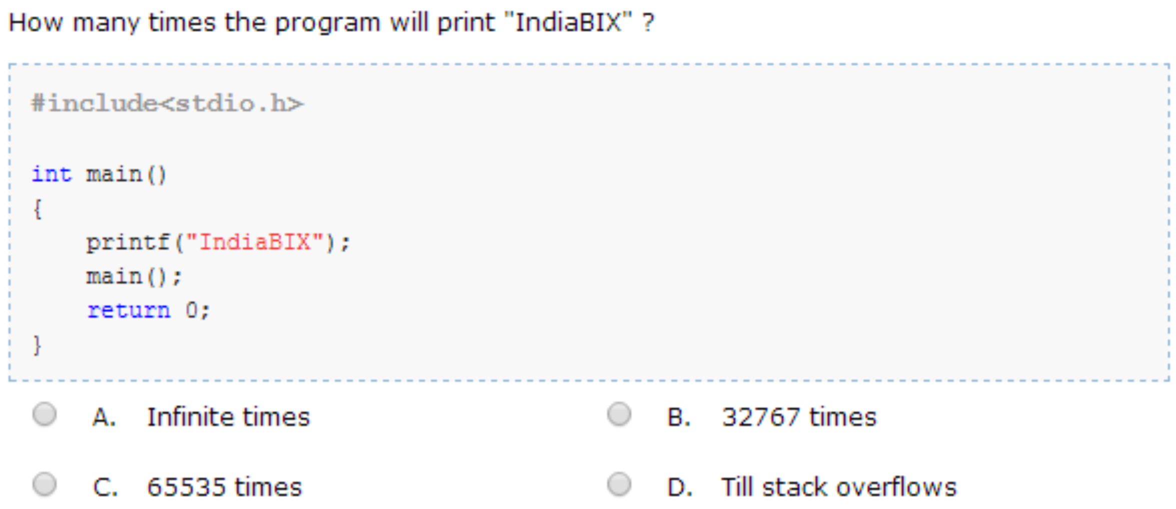
○ C. 4, 4

○ D. Output may vary from compiler to compiler



In the expression `a=b=5` the order of Assignment is NOT decided by Associativity of operators

○ A. True ○ B. False



How many times the program will print "IndiaBIX" ?

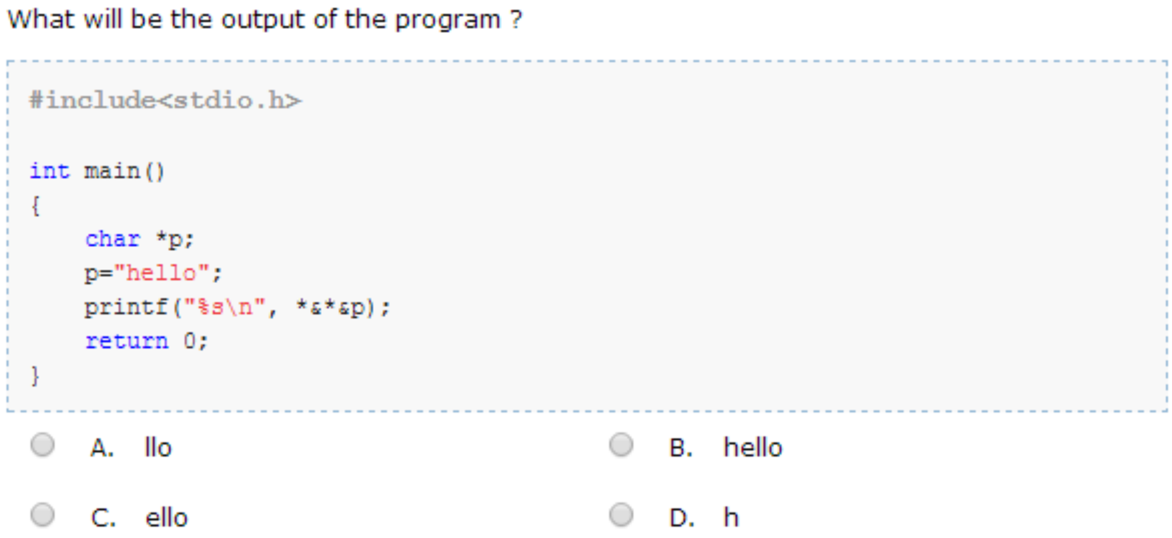
```
#include<stdio.h>

int main()
{
    printf("IndiaBIX");
    main();
    return 0;
}
```

○ A. Infinite times

○ B. 32767 times

○ C. 65535 times

○ D. Till stack overflows



What will be the output of the program ?

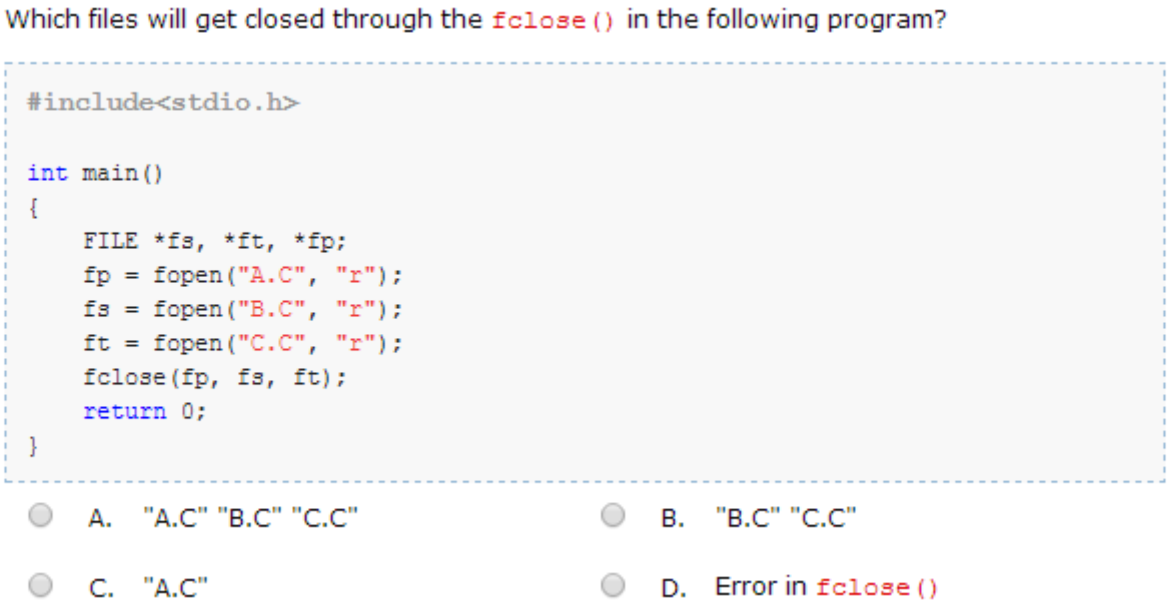
```
#include<stdio.h>

int main()
{
    char *p;
    p="hello";
    printf("%s\n", *&*&p);
    return 0;
}
```

○ A. llo

○ B. hello

○ C. ello

○ D. h



Which files will get closed through the `fclose()` in the following program?

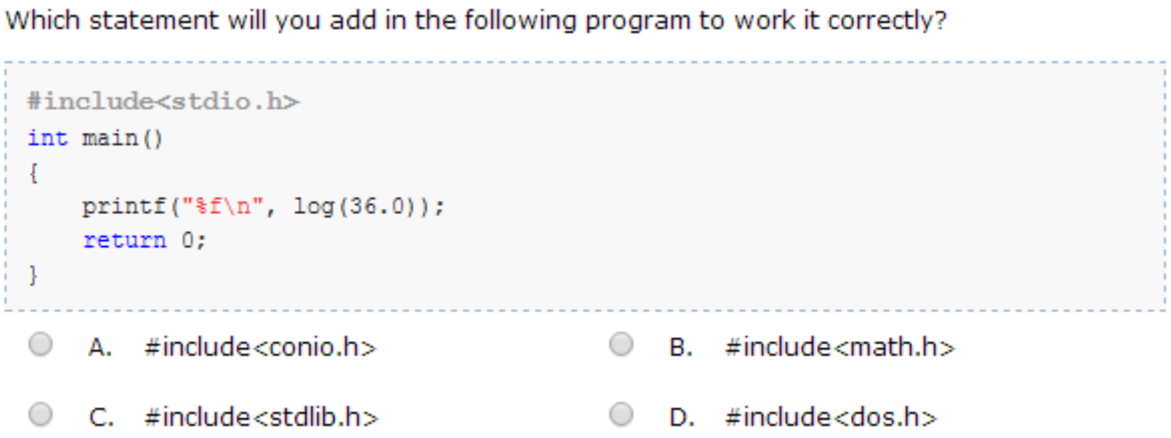
```
#include<stdio.h>

int main()
{
    FILE *fs, *ft, *fp;
    fp = fopen("A.C", "r");
    fs = fopen("B.C", "r");
    ft = fopen("C.C", "r");
    fclose(fp, fs, ft);
    return 0;
}
```

○ A. "A.C" "B.C" "C.C"

○ B. "B.C" "C.C"

○ C. "A.C"

○ D. Error in `fclose()`



Which statement will you add in the following program to work it correctly?

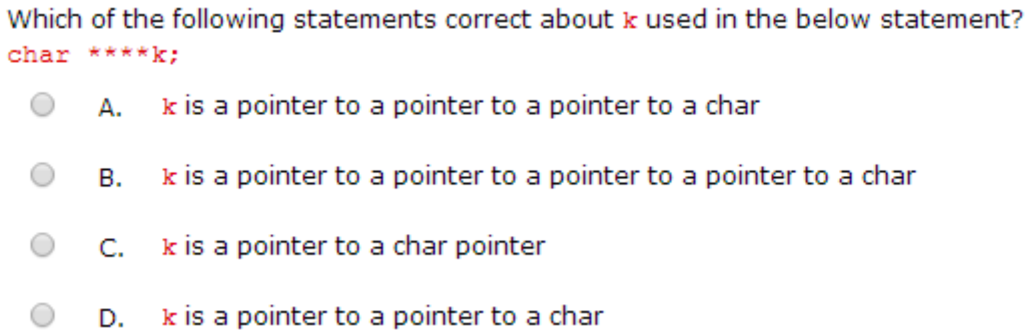
```
#include<stdio.h>
int main()
{
    printf("%f\n", log(36.0));
    return 0;
}
```

○ A. #include<conio.h>

○ B. #include<math.h>

○ C. #include<stdlib.h>

○ D. #include<dos.h>



Which of the following statements correct about `k` used in the below statement?

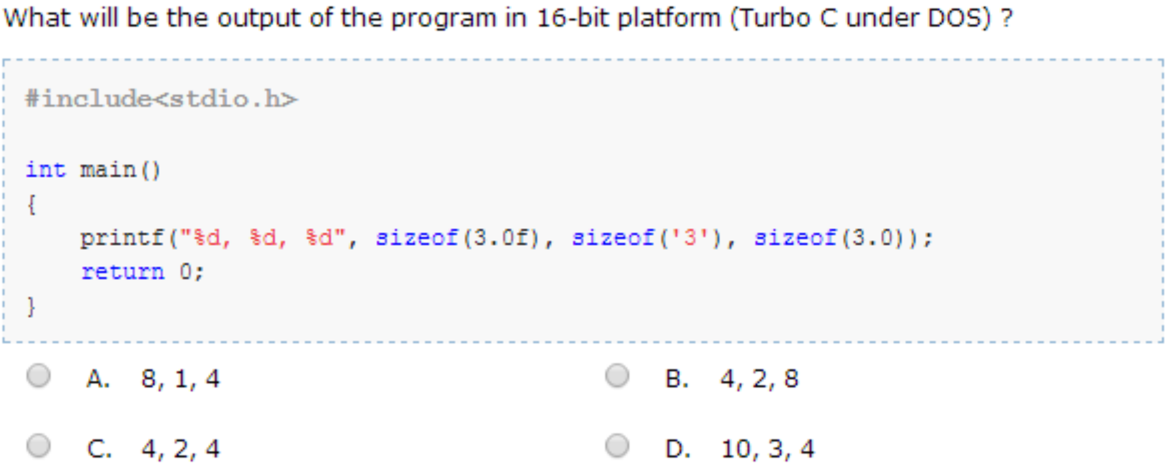
```
char ****k;
```

○ A. `k` is a pointer to a pointer to a pointer to a char

○ B. `k` is a pointer to a pointer to a pointer to a pointer to a char

○ C. `k` is a pointer to a char pointer

○ D. `k` is a pointer to a pointer to a char

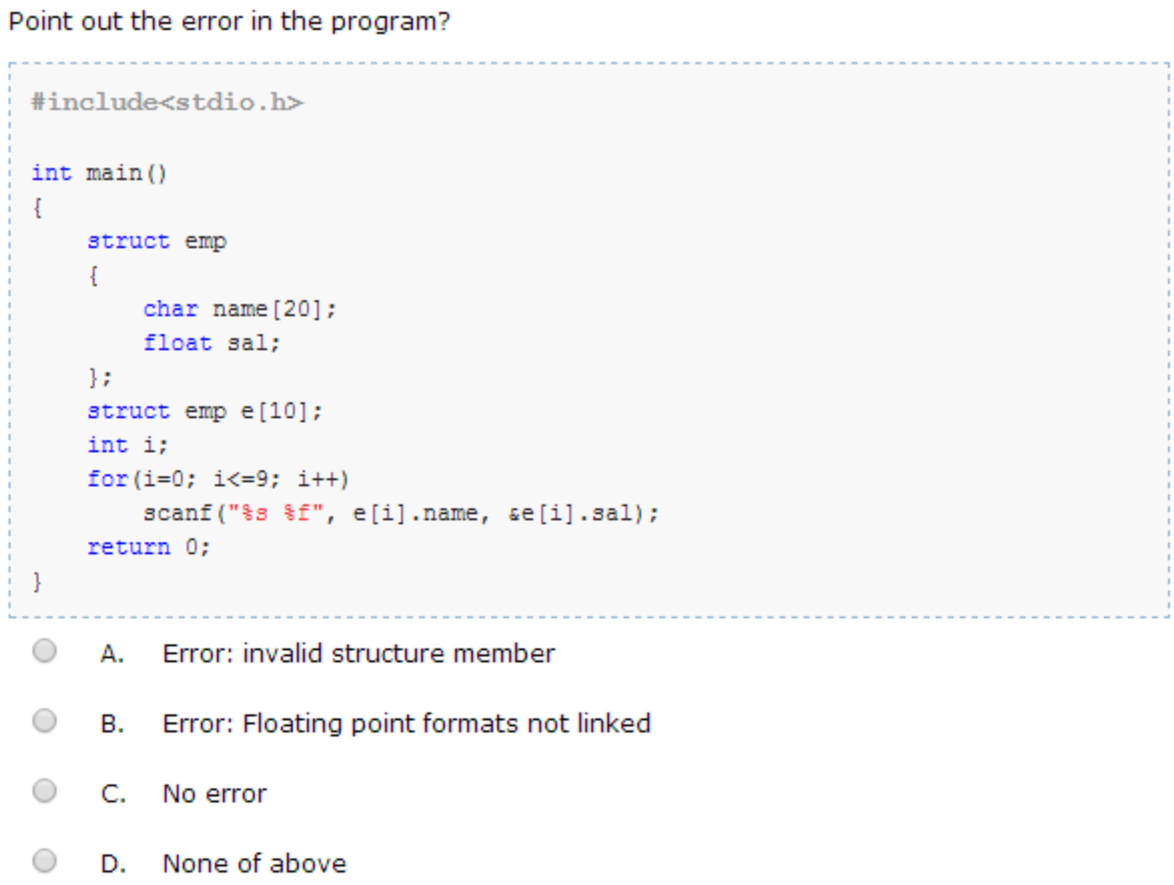


What will be the output of the program in 16-bit platform (Turbo C under DOS) ?

```
#include<stdio.h>

int main()
{
    printf("%d, %d, %d", sizeof(3.0f), sizeof('3'), sizeof(3.0));
    return 0;
}
```

○ A. 8, 1, 4 ○ B. 4, 2, 8

○ C. 4, 2, 4 ○ D. 10, 3, 4



Point out the error in the program?

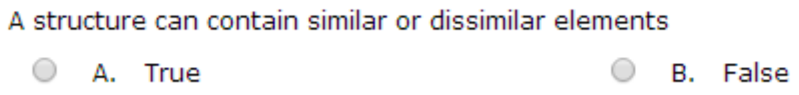
```
#include<stdio.h>

int main()
{
    struct emp
    {
        char name[20];
        float sal;
    };
    struct emp e[10];
    int i;
    for(i=0; i<=9; i++)
        scanf("%s %f", e[i].name, &e[i].sal);
    return 0;
}
```

○ A. Error: invalid structure member

○ B. Error: Floating point formats not linked

○ C. No error

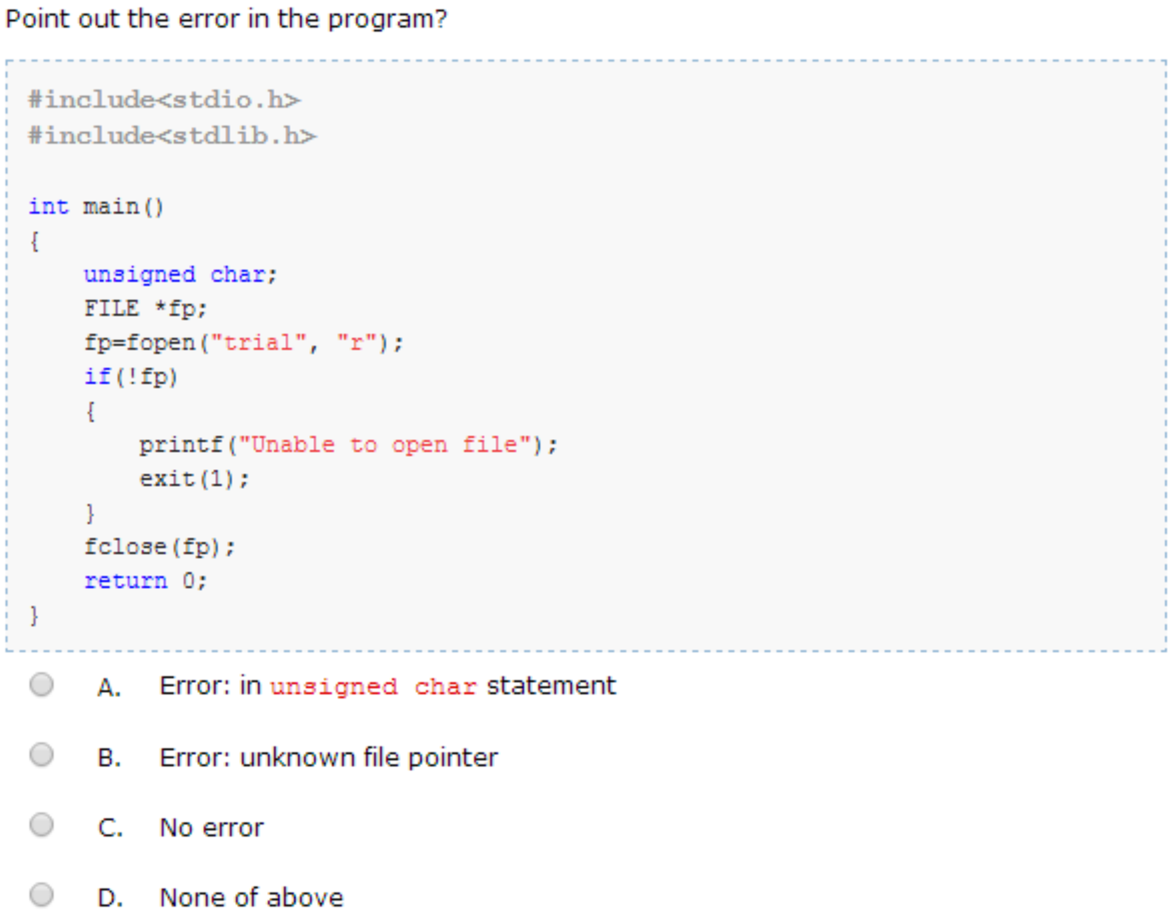
○ D. None of above



A structure can contain similar or dissimilar elements

○ A. True

○ B. False



Point out the error in the program?

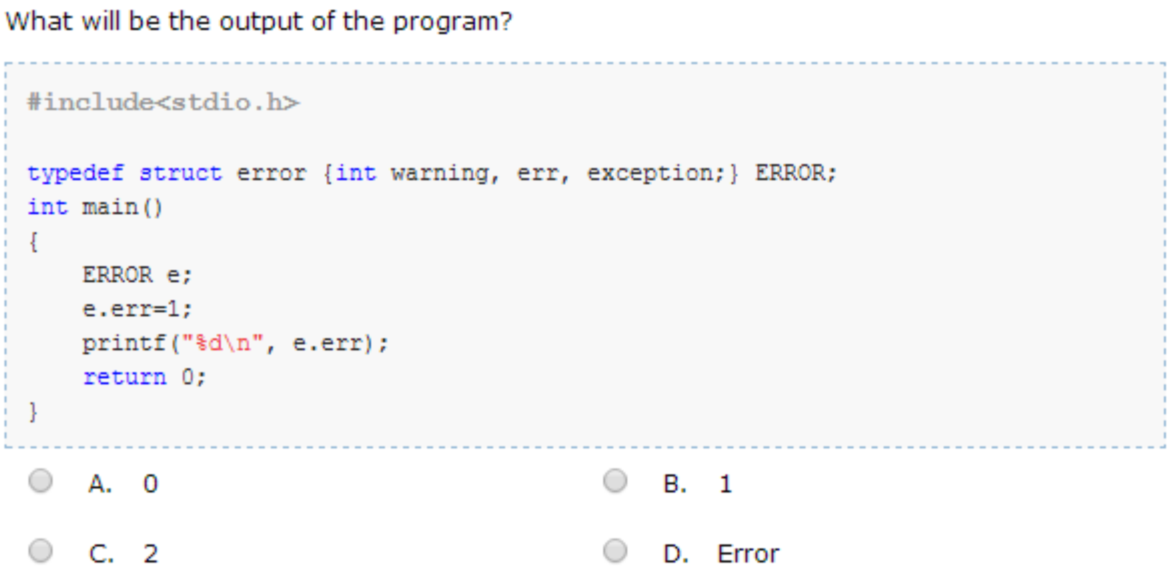
```
#include<stdio.h>
#include<stdlib.h>

int main()
{
    unsigned char;
    FILE *fp;
    fp=fopen("trial", "r");
    if(!fp)
    {
        printf("Unable to open file");
        exit(1);
    }
    fclose(fp);
    return 0;
}
```

○ A. Error: in `unsigned char` statement

○ B. Error: unknown file pointer

○ C. No error

○ D. None of above



What will be the output of the program?

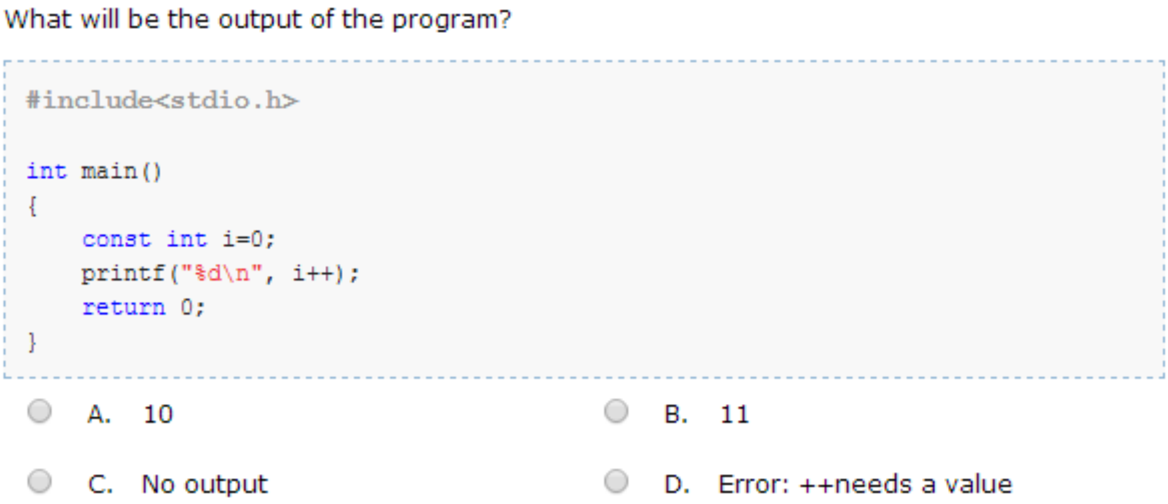
```
#include<stdio.h>

typedef struct error {int warning, err, exception;} ERROR;
int main()
{
    ERROR e;
    e.err=1;
    printf("%d\n", e.err);
    return 0;
}
```

○ A. 0

○ B. 1

○ C. 2

○ D. Error

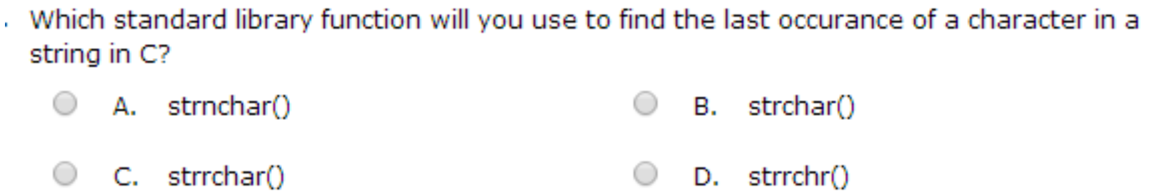


What will be the output of the program?

```
#include<stdio.h>

int main()
{
    const int i=0;
    printf("%d\n", i++);
    return 0;
}
```

○ A. 10　○ B. 11

○ C. No output　○ D. Error: ++needs a value



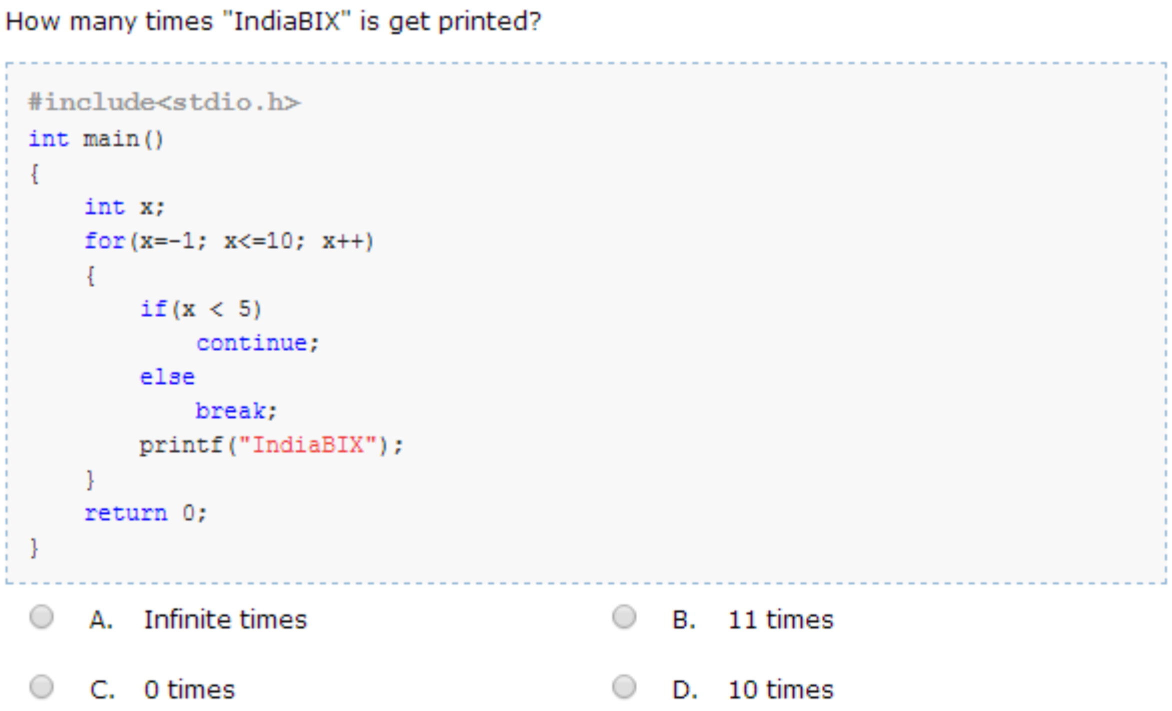
Which standard library function will you use to find the last occurance of a character in a string in C?

○ A. strnchar()

○ B. strchar()

○ C. strrchar()

○ D. strrchr()

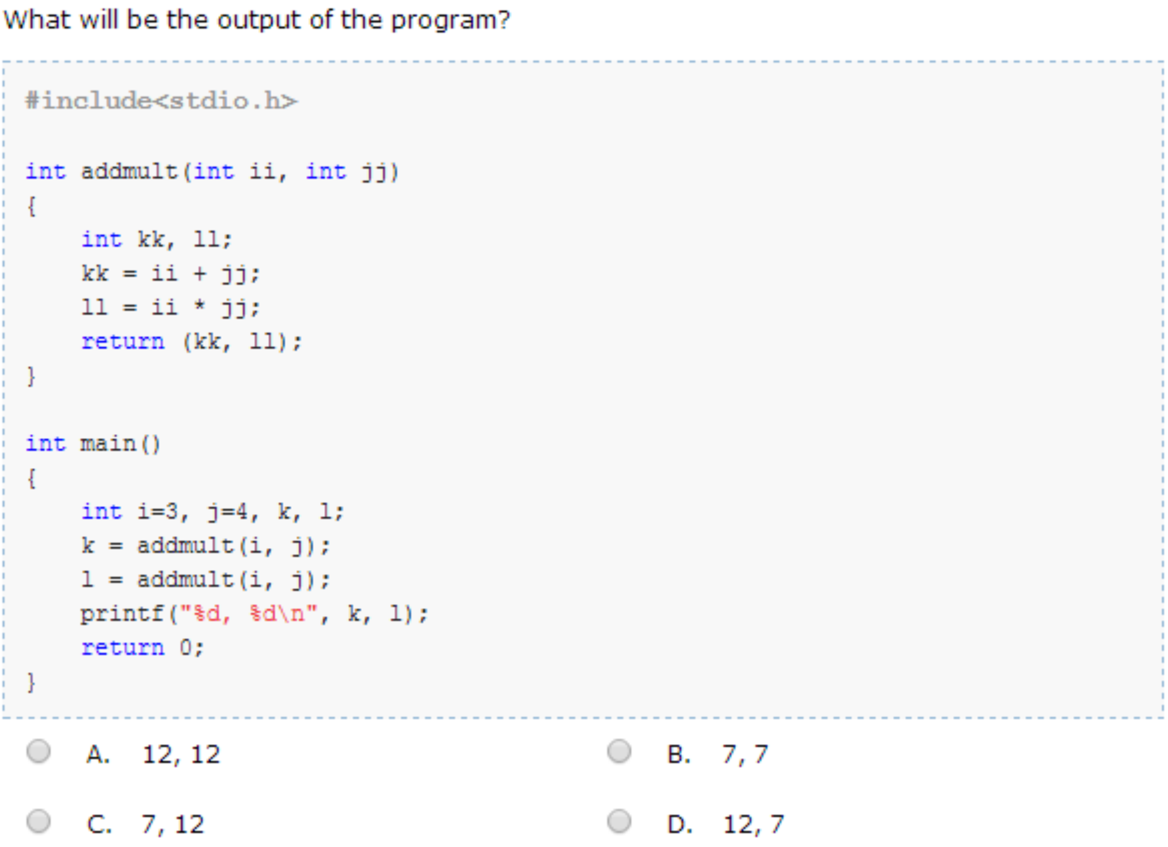


How many times "IndiaBIX" is get printed?

```
#include<stdio.h>
int main()
{
    int x;
    for(x=-1; x<=10; x++)
    {
        if(x < 5)
            continue;
        else
            break;
        printf("IndiaBIX");
    }
    return 0;
}
```

○ A. Infinite times ○ B. 11 times

○ C. 0 times ○ D. 10 times

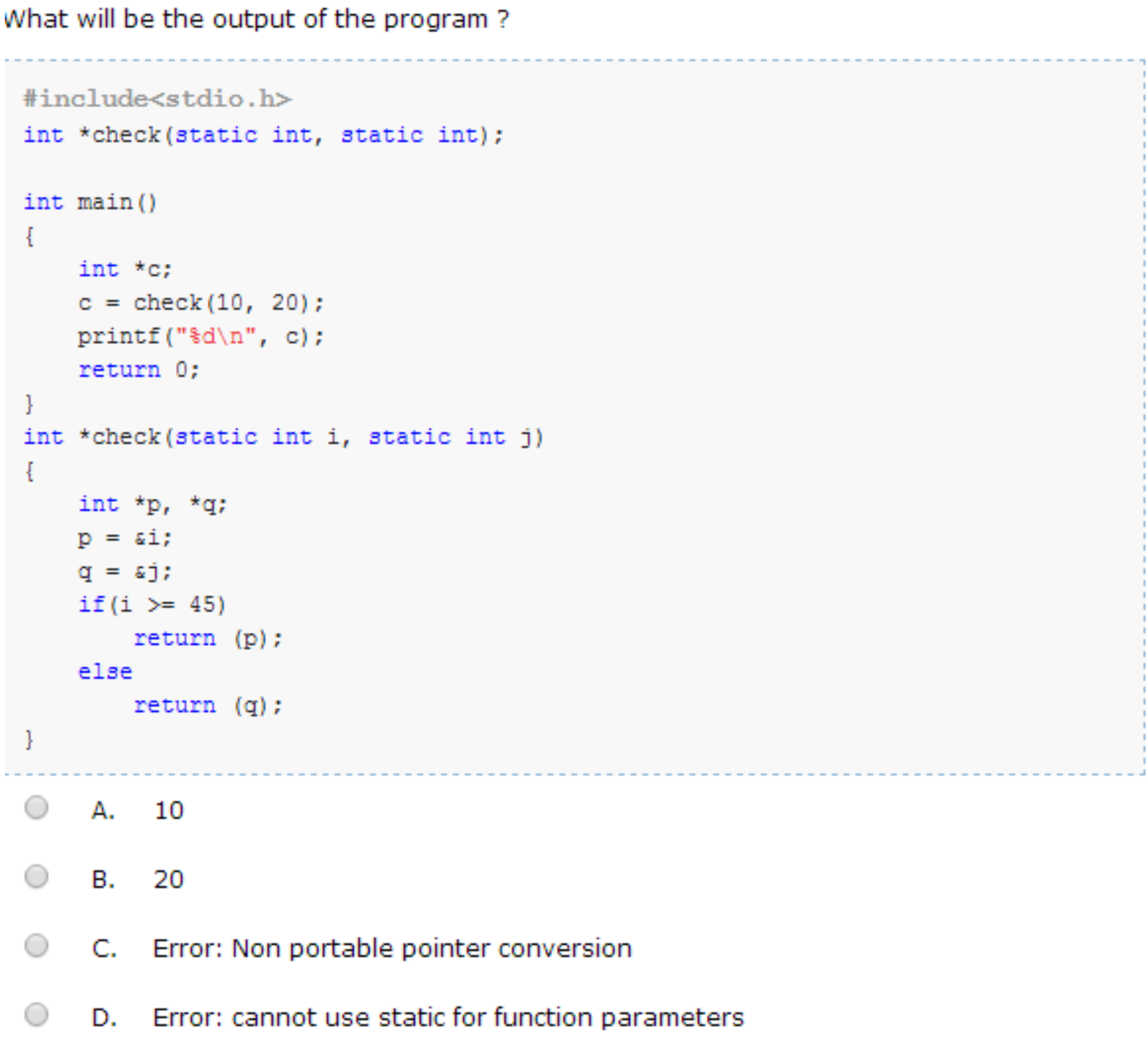


What will be the output of the program?

```c
#include<stdio.h>

int addmult(int ii, int jj)
{
    int kk, ll;
    kk = ii + jj;
    ll = ii * jj;
    return (kk, ll);
}

int main()
{
    int i=3, j=4, k, l;
    k = addmult(i, j);
    l = addmult(i, j);
    printf("%d, %d\n", k, l);
    return 0;
}
```

○ A. 12, 12 ○ B. 7, 7

○ C. 7, 12 ○ D. 12, 7



What will be the output of the program ?

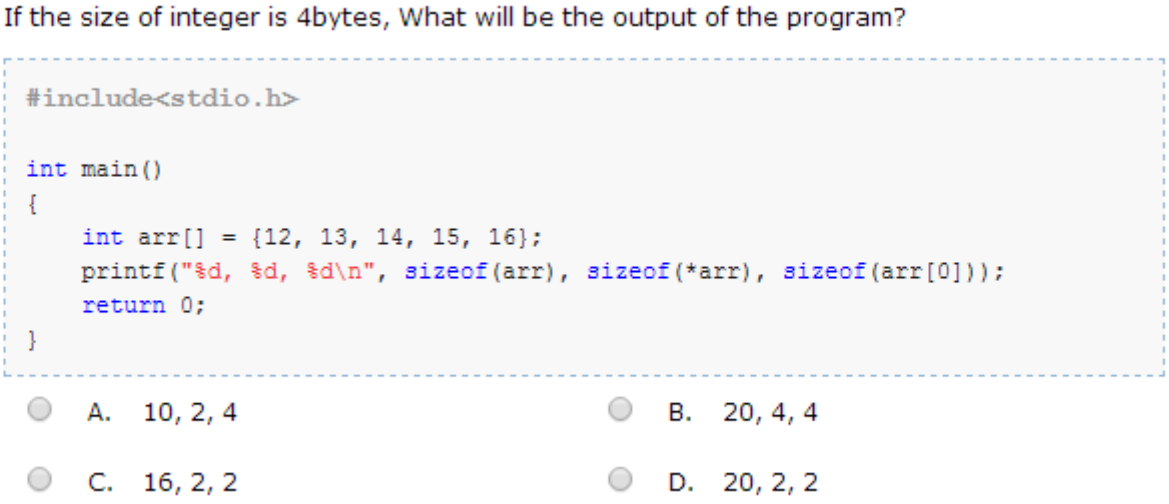
```
#include<stdio.h>
int *check(static int, static int);

int main()
{
    int *c;
    c = check(10, 20);
    printf("%d\n", c);
    return 0;
}
int *check(static int i, static int j)
{
    int *p, *q;
    p = &i;
    q = &j;
    if(i >= 45)
        return (p);
    else
        return (q);
}
```

○ A. 10

○ B. 20

○ C. Error: Non portable pointer conversion

○ D. Error: cannot use static for function parameters



If the size of integer is 4bytes, What will be the output of the program?

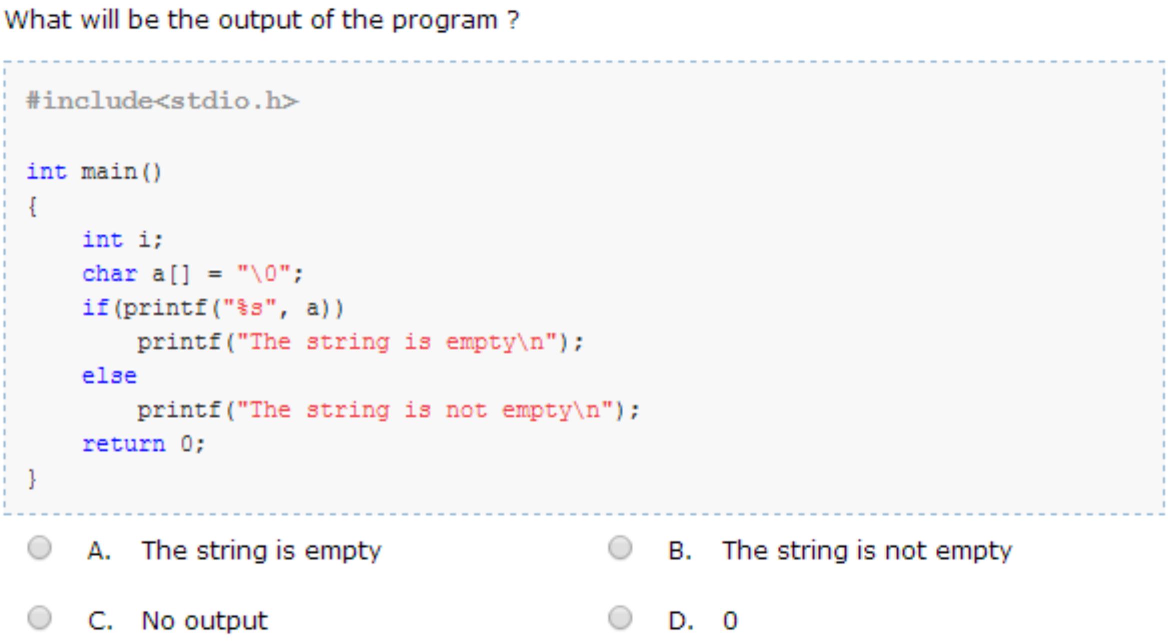
```
#include<stdio.h>

int main()
{
    int arr[] = {12, 13, 14, 15, 16};
    printf("%d, %d, %d\n", sizeof(arr), sizeof(*arr), sizeof(arr[0]));
    return 0;
}
```

○ A. 10, 2, 4

○ B. 20, 4, 4

○ C. 16, 2, 2

○ D. 20, 2, 2

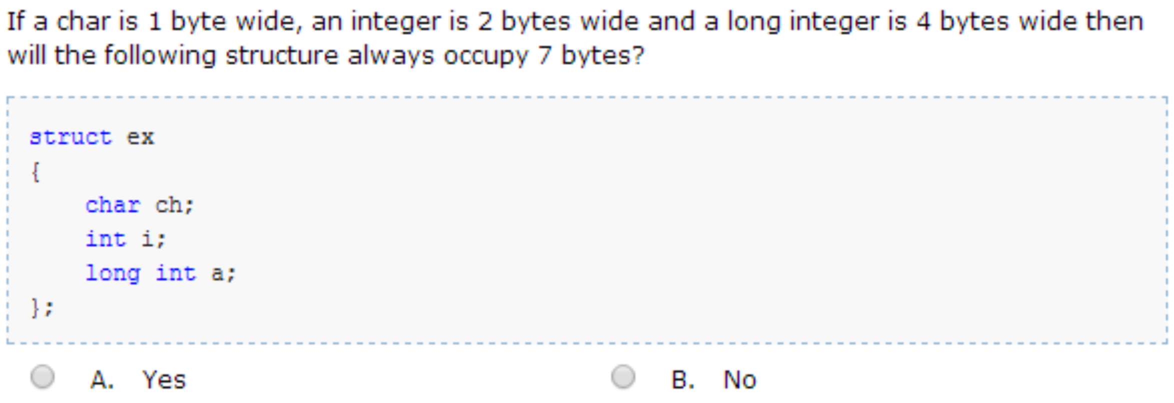


What will be the output of the program ?

```
#include<stdio.h>

int main()
{
    int i;
    char a[] = "\0";
    if(printf("%s", a))
        printf("The string is empty\n");
    else
        printf("The string is not empty\n");
    return 0;
}
```

○ A. The string is empty ○ B. The string is not empty

○ C. No output ○ D. 0

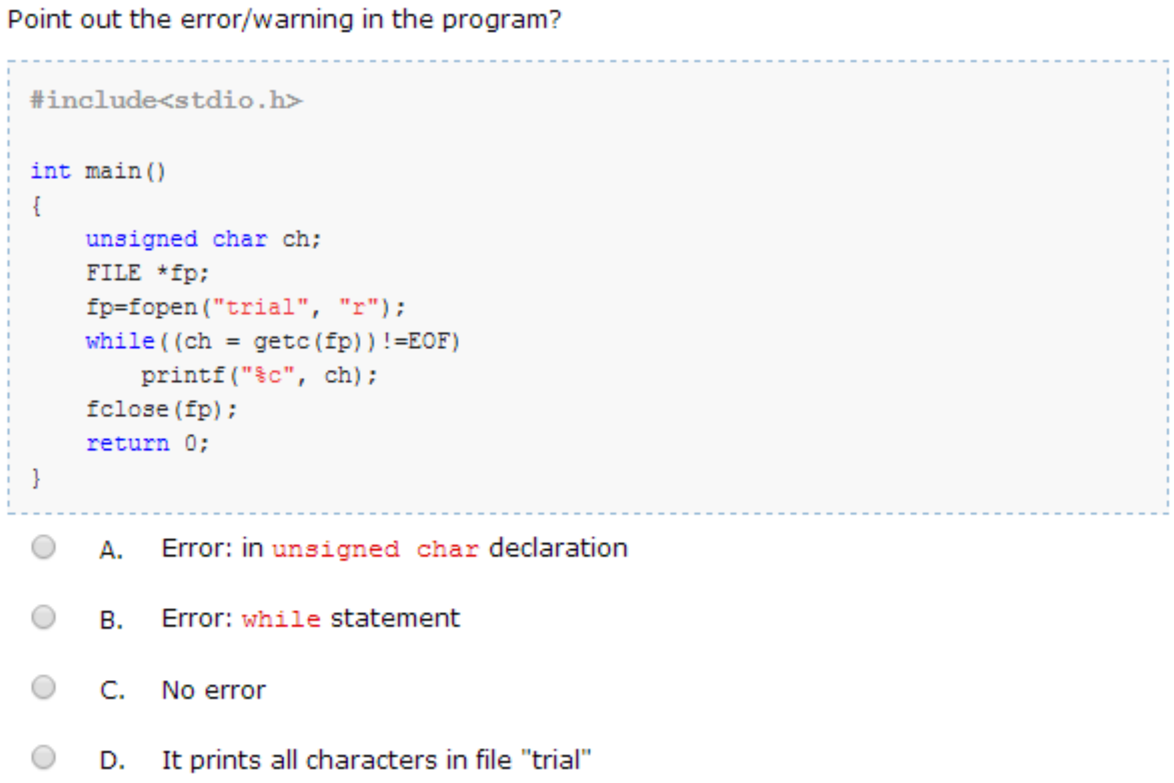


If a char is 1 byte wide, an integer is 2 bytes wide and a long integer is 4 bytes wide then will the following structure always occupy 7 bytes?

```
struct ex
{
    char ch;
    int i;
    long int a;
};
```

○ A. Yes

○ B. No



Point out the error/warning in the program?

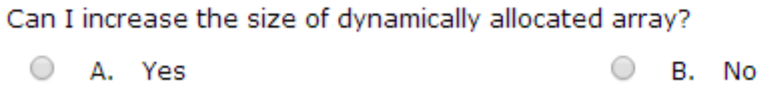
```
#include<stdio.h>

int main()
{
    unsigned char ch;
    FILE *fp;
    fp=fopen("trial", "r");
    while((ch = getc(fp))!=EOF)
        printf("%c", ch);
    fclose(fp);
    return 0;
}
```

○ A. Error: in `unsigned char` declaration

○ B. Error: `while` statement

○ C. No error

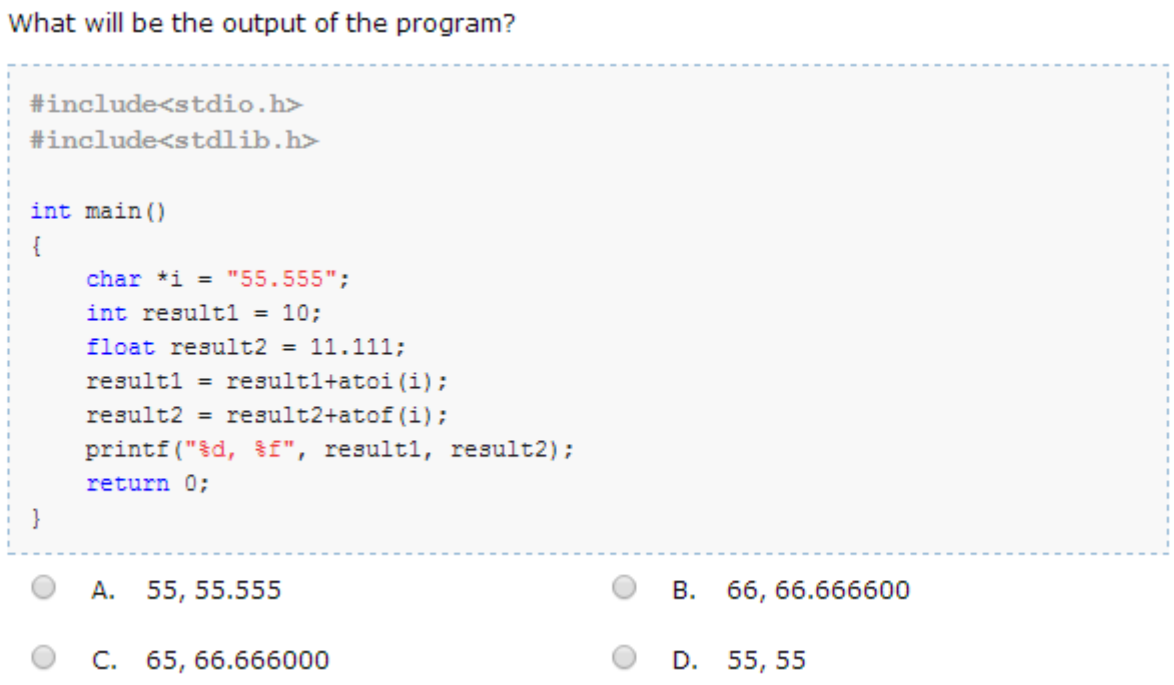
○ D. It prints all characters in file "trial"



Can I increase the size of dynamically allocated array?

○ A. Yes

○ B. No



What will be the output of the program?

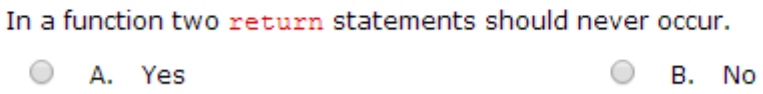
```
#include<stdio.h>
#include<stdlib.h>

int main()
{
    char *i = "55.555";
    int result1 = 10;
    float result2 = 11.111;
    result1 = result1+atoi(i);
    result2 = result2+atof(i);
    printf("%d, %f", result1, result2);
    return 0;
}
```

○ A. 55, 55.555

○ B. 66, 66.666600

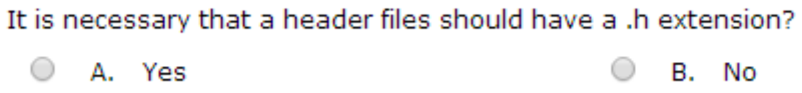
○ C. 65, 66.666000

○ D. 55, 55



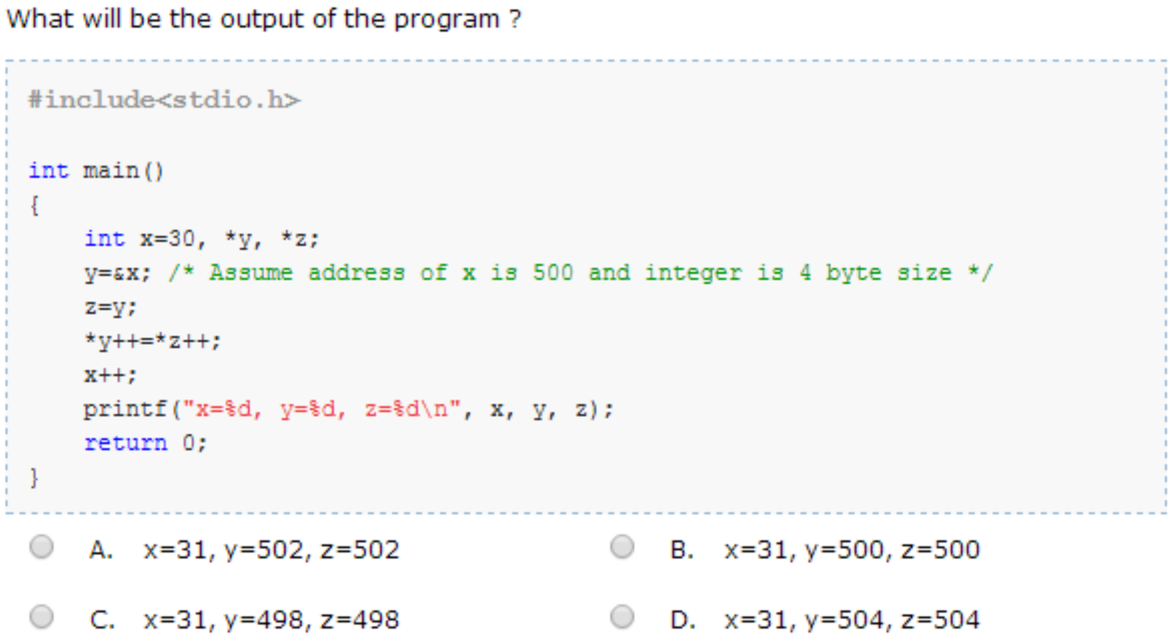
In a function two `return` statements should never occur.

○ A. Yes ○ B. No



It is necessary that a header files should have a .h extension?

○ A. Yes ○ B. No

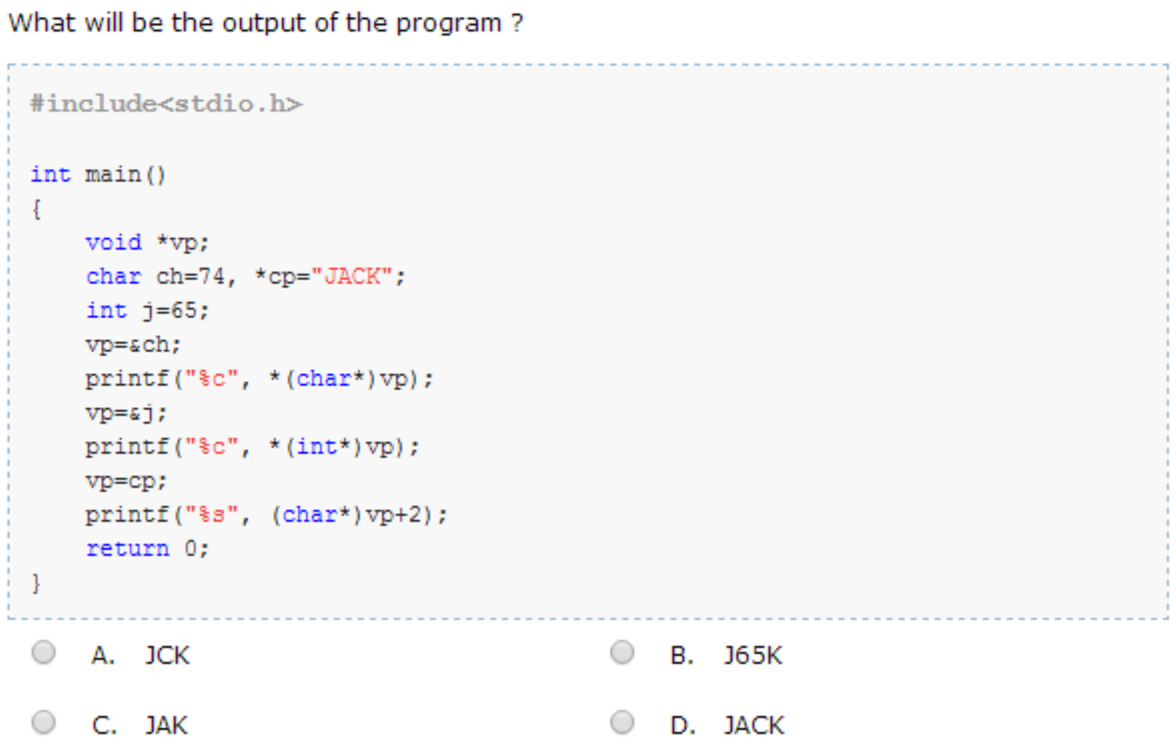


What will be the output of the program ?

```
#include<stdio.h>

int main()
{
    int x=30, *y, *z;
    y=&x; /* Assume address of x is 500 and integer is 4 byte size */
    z=y;
    *y++=*z++;
    x++;
    printf("x=%d, y=%d, z=%d\n", x, y, z);
    return 0;
}
```

○ A. x=31, y=502, z=502  ○ B. x=31, y=500, z=500

○ C. x=31, y=498, z=498  ○ D. x=31, y=504, z=504

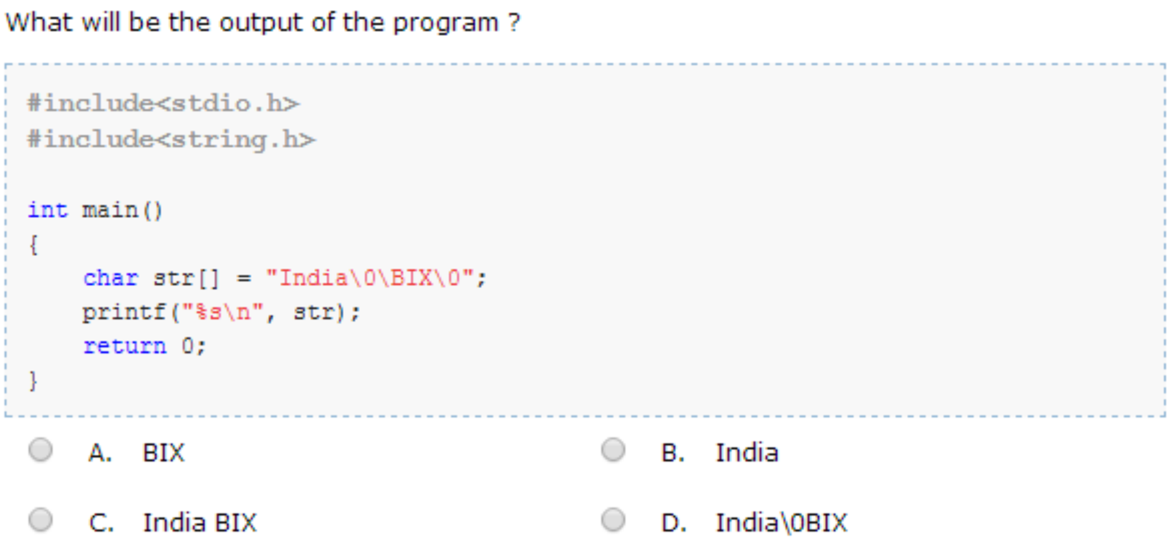


What will be the output of the program ?

```
#include<stdio.h>

int main()
{
    void *vp;
    char ch=74, *cp="JACK";
    int j=65;
    vp=&ch;
    printf("%c", *(char*)vp);
    vp=&j;
    printf("%c", *(int*)vp);
    vp=cp;
    printf("%s", (char*)vp+2);
    return 0;
}
```

○ A. JCK ○ B. J65K

○ C. JAK ○ D. JACK



What will be the output of the program ?

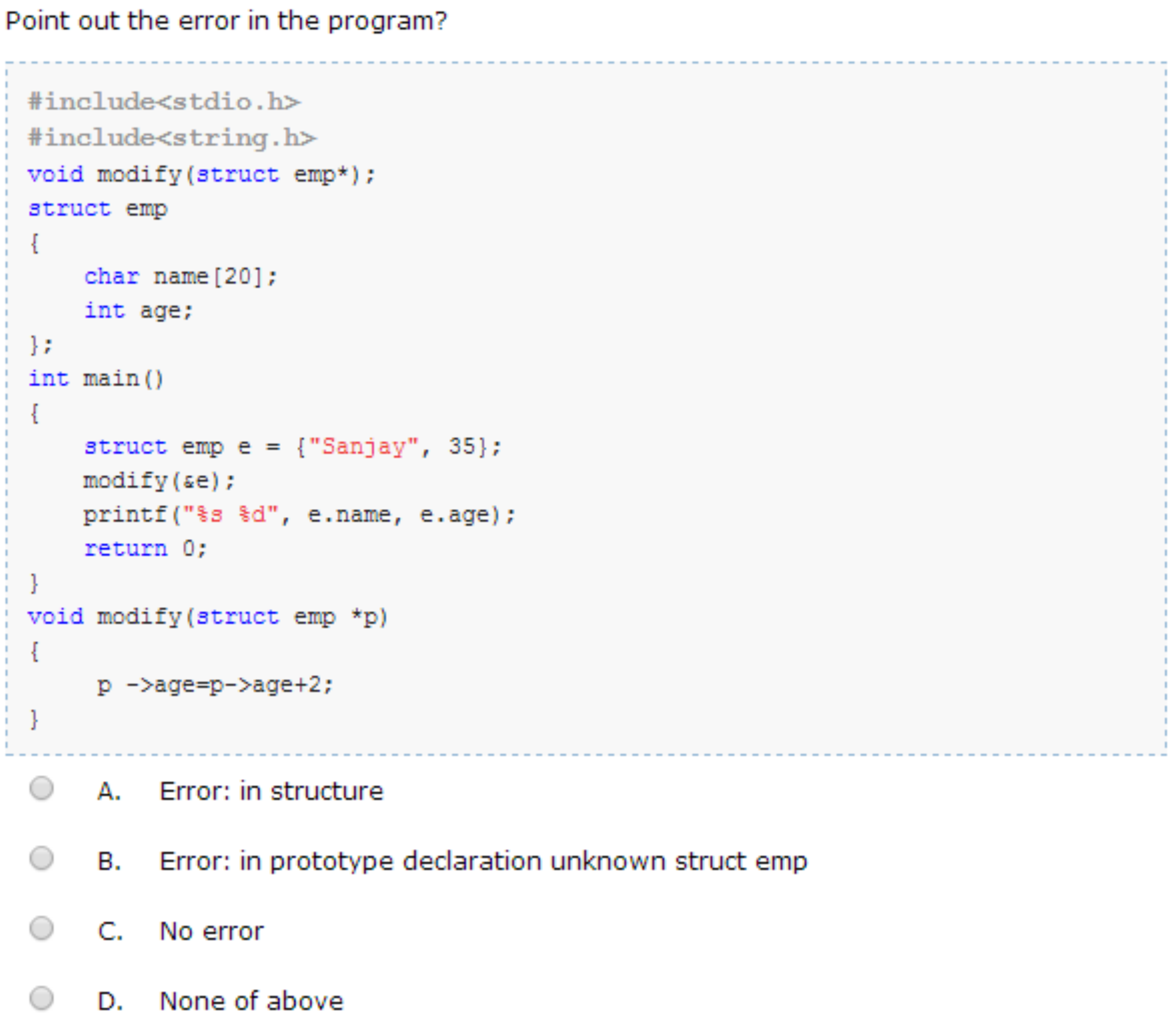
```
#include<stdio.h>
#include<string.h>

int main()
{
    char str[] = "India\0\BIX\0";
    printf("%s\n", str);
    return 0;
}
```

○ A. BIX

○ B. India

○ C. India BIX

○ D. India\0BIX



Point out the error in the program?

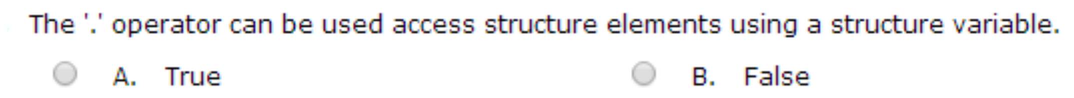
```
#include<stdio.h>
#include<string.h>
void modify(struct emp*);
struct emp
{
    char name[20];
    int age;
};
int main()
{
    struct emp e = {"Sanjay", 35};
    modify(&e);
    printf("%s %d", e.name, e.age);
    return 0;
}
void modify(struct emp *p)
{
     p ->age=p->age+2;
}
```

○ A. Error: in structure

○ B. Error: in prototype declaration unknown struct emp

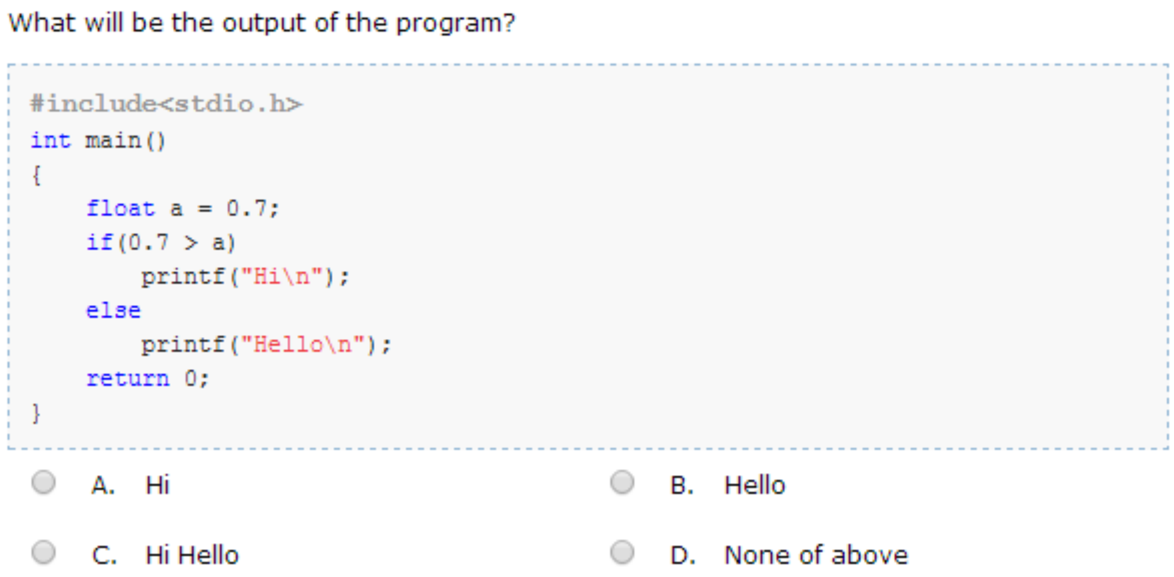
○ C. No error

○ D. None of above



The '.' operator can be used access structure elements using a structure variable.

○ A. True ○ B. False



What will be the output of the program?

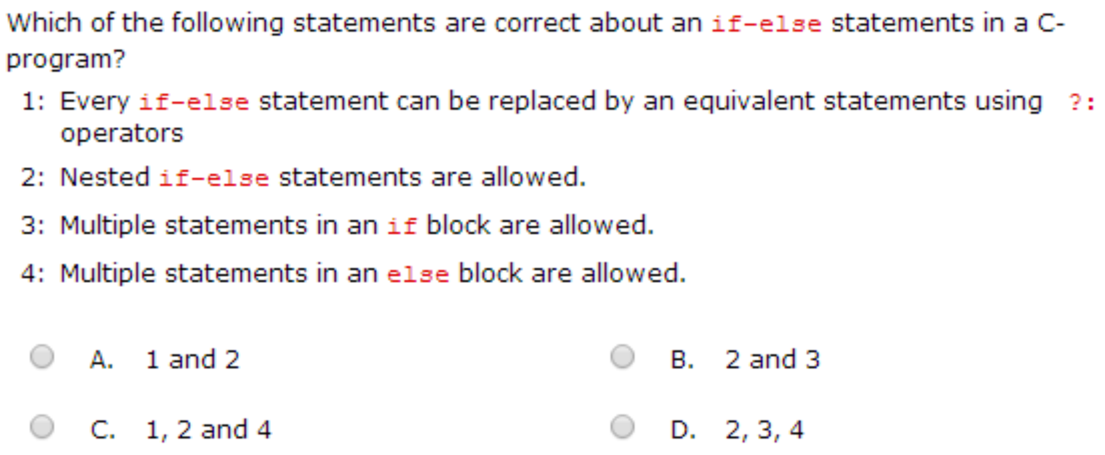
```
#include<stdio.h>
int main()
{
    float a = 0.7;
    if(0.7 > a)
        printf("Hi\n");
    else
        printf("Hello\n");
    return 0;
}
```

○ A. Hi

○ B. Hello

○ C. Hi Hello

○ D. None of above



Which of the following statements are correct about an `if-else` statements in a C-program?

1: Every `if-else` statement can be replaced by an equivalent statements using `?:` operators

2: Nested `if-else` statements are allowed.

3: Multiple statements in an `if` block are allowed.

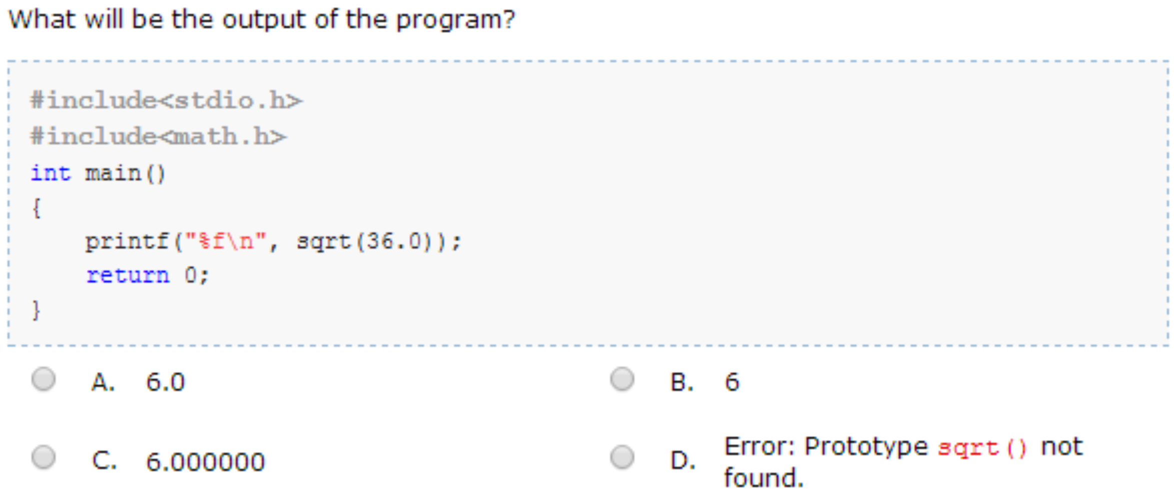
4: Multiple statements in an `else` block are allowed.

○ A. 1 and 2

○ B. 2 and 3

○ C. 1, 2 and 4

○ D. 2, 3, 4



What will be the output of the program?

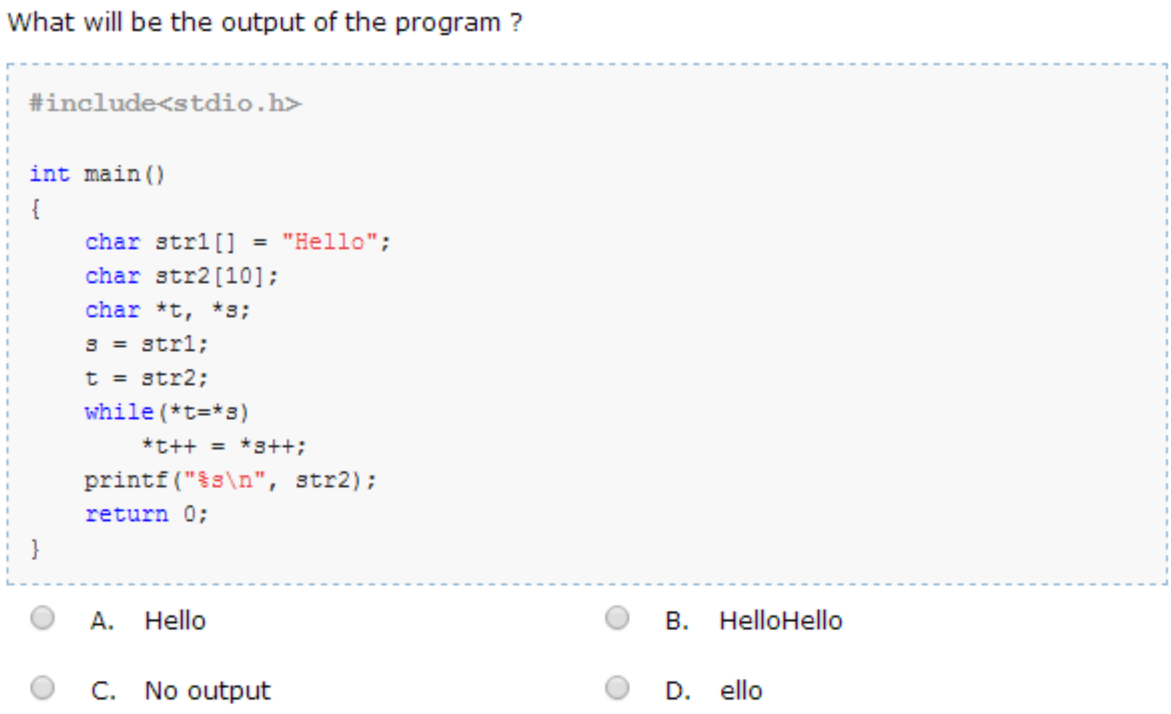
```
#include<stdio.h>
#include<math.h>
int main()
{
    printf("%f\n", sqrt(36.0));
    return 0;
}
```

○ A. 6.0

○ B. 6

○ C. 6.000000

○ D. Error: Prototype `sqrt()` not found.



What will be the output of the program ?

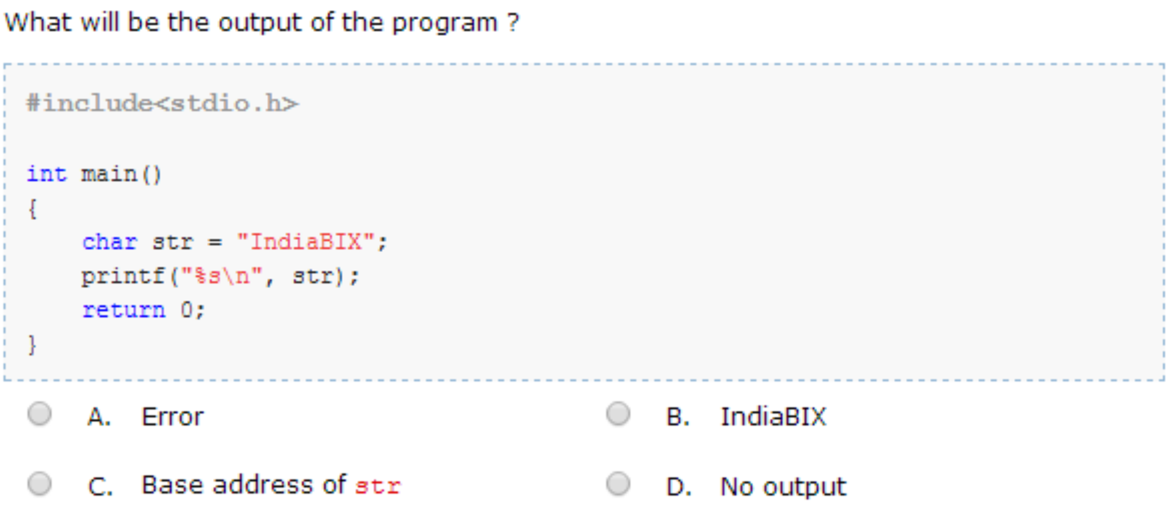
```
#include<stdio.h>

int main()
{
    char str1[] = "Hello";
    char str2[10];
    char *t, *s;
    s = str1;
    t = str2;
    while(*t=*s)
        *t++ = *s++;
    printf("%s\n", str2);
    return 0;
}
```

○ A. Hello

○ B. HelloHello

○ C. No output

○ D. ello



What will be the output of the program ?

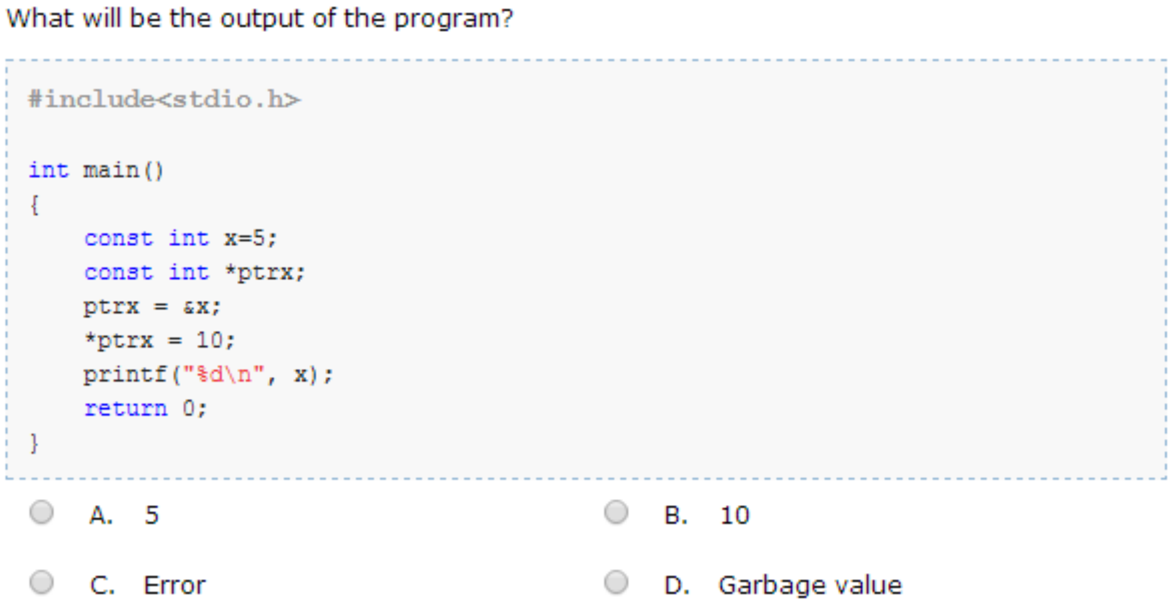
```
#include<stdio.h>

int main()
{
    char str = "IndiaBIX";
    printf("%s\n", str);
    return 0;
}
```

○ A. Error

○ B. IndiaBIX

○ C. Base address of `str`

○ D. No output

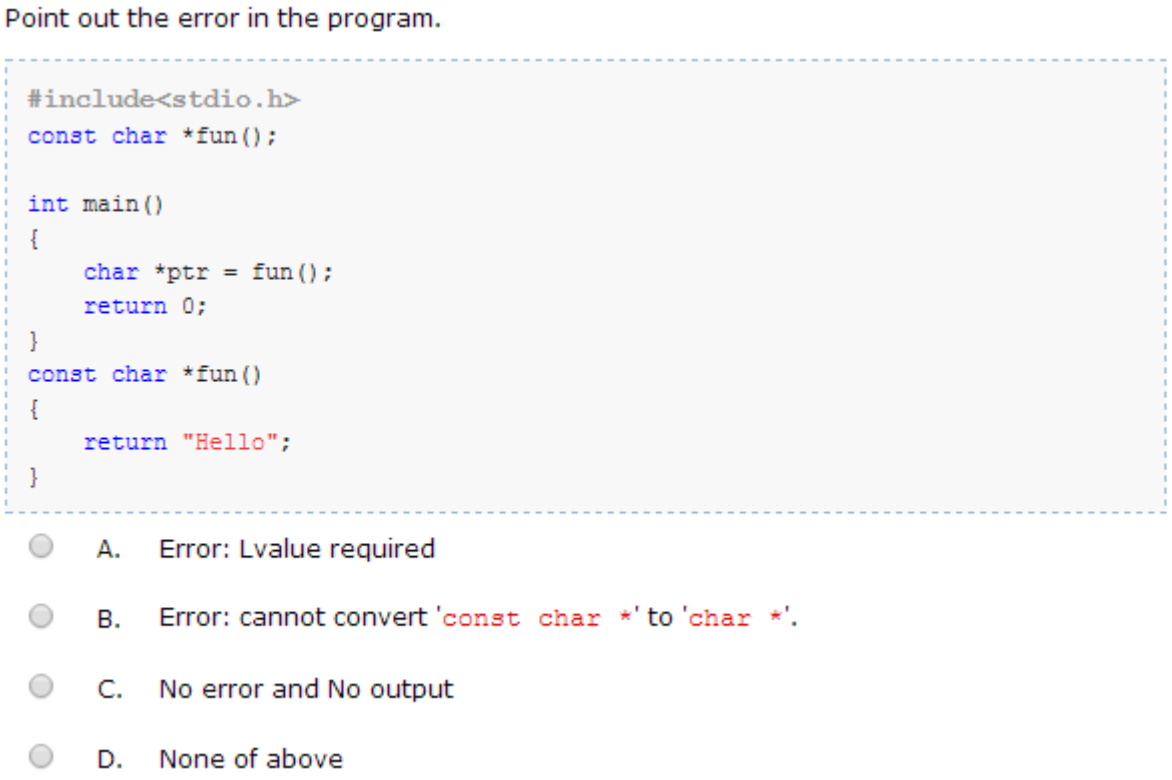


What will be the output of the program?

```
#include<stdio.h>

int main()
{
    const int x=5;
    const int *ptrx;
    ptrx = &x;
    *ptrx = 10;
    printf("%d\n", x);
    return 0;
}
```

○ A. 5 ○ B. 10

○ C. Error ○ D. Garbage value



Point out the error in the program.

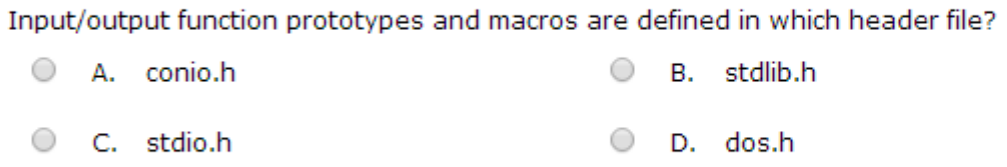
```
#include<stdio.h>
const char *fun();

int main()
{
    char *ptr = fun();
    return 0;
}
const char *fun()
{
    return "Hello";
}
```

○ A. Error: Lvalue required

○ B. Error: cannot convert 'const char *' to 'char *'.

○ C. No error and No output

○ D. None of above



Input/output function prototypes and macros are defined in which header file?

○ A. conio.h

○ B. stdlib.h

○ C. stdio.h

○ D. dos.h

1. What is the correct value to return to the operating system upon the successful completion of a program?
A. -1
B. 1
C. 0
D. Programs do not return a value.

2. What is the only function all C programs must contain?
A. start()
B. system()
C. main()
D. program()

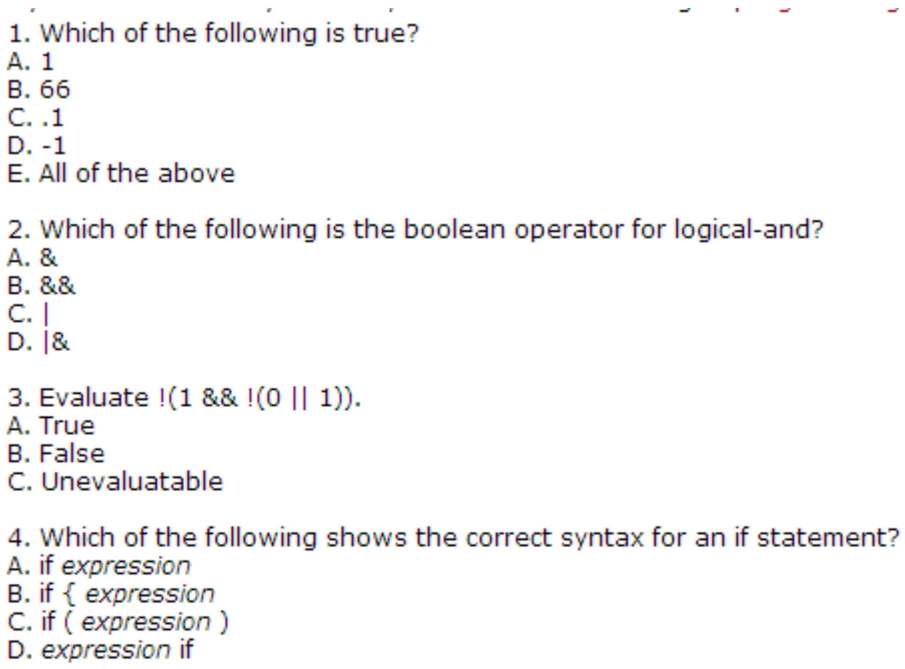
3. What punctuation is used to signal the beginning and end of code blocks?
A. { }
B. -> and <-
C. BEGIN and END
D. ( and )

4. What punctuation ends most lines of C code?
A. .
B. ;
C. :
D. '

5. Which of the following is a correct comment?
A. */ Comments */
B. ** Comment **
C. /* Comment */
D. { Comment }

6. Which of the following is not a correct variable type?
A. float
B. real
C. int
D. double

7. Which of the following is the correct operator to compare two variables?
A. :=
B. =
C. equal
D. ==



1. Which of the following is true?
A. 1
B. 66
C. .1
D. -1
E. All of the above

2. Which of the following is the boolean operator for logical-and?
A. &
B. &&
C. |
D. |&

3. Evaluate !(1 && !(0 || 1)).
A. True
B. False
C. Unevaluatable

4. Which of the following shows the correct syntax for an if statement?
A. if *expression*
B. if { *expression*
C. if ( *expression* )
D. *expression* if

1. What is the final value of x when the code int x; for(x=0; x<10; x++) {} is run?
A. 10
B. 9
C. 0
D. 1

2. When does the code block following while(x<100) execute?
A. When x is less than one hundred
B. When x is greater than one hundred
C. When x is equal to one hundred
D. While it wishes

3. Which is not a loop structure?
A. for
B. do while
C. while
D. repeat until

4. How many times is a do while loop guaranteed to loop?
A. 0
B. Infinitely
C. 1
D. Variable

1. Which is not a proper prototype?
A. int funct(char x, char y);
B. double funct(char x)
C. void funct();
D. char x();

2. What is the return type of the function with prototype: "int func(char x, float v, double t);"
A. char
B. int
C. float
D. double

3. Which of the following is a valid function call (assuming the function exists)?
A. funct;
B. funct x, y;
C. funct();
D. int funct();

4. Which of the following is a complete function?
A. int funct();
B. int funct(int x) {return x=x+1;}
C. void funct(int) {printf( "Hello" );
D. void funct(x) {printf( "Hello" ); }

1. Which follows the case statement?
A. :
B. ;
C. -
D. A newline

2. What is required to avoid falling through from one case to the next?
A. end;
B. break;
C. Stop;
D. A semicolon.

3. What keyword covers unhandled possibilities?
A. all
B. contingency
C. default
D. other

4. What is the result of the following code?

```
int x=0;

switch(x)

{

  case 1: printf( "One" );

  case 0: printf( "Zero" );

  case 2: printf( "Hello World" );

}
```

A. One
B. Zero
C. Hello World
D. ZeroHello World

1. Which of the following is the proper declaration of a pointer?
A. int x;
B. int &x;
C. ptr x;
D. int *x;

2. Which of the following gives the memory address of integer variable a?
A. *a;
B. a;
C. &a;
D. address(a);

3. Which of the following gives the memory address of a variable pointed to by pointer a?
A. a;
B. *a;
C. &a;
D. address(a);

4. Which of the following gives the value stored at the address pointed to by pointer a?
A. a;
B. val(a);
C. *a;
D. &a;

1. Which of the following accesses a variable in structure b?
A. b->var;
B. b.var;
C. b-var;
D. b>var;

2. Which of the following accesses a variable in a pointer to a structure, *b?
A. b->var;
B. b.var;
C. b-var;
D. b>var;

3. Which of the following is a properly defined struct?
A. struct {int a;}
B. struct a_struct {int a;}
C. struct a_struct int a;
D. struct a_struct {int a;};

4. Which properly declares a variable of struct foo?
A. struct foo;
B. struct foo var;
C. foo;
D. int foo;

1. Which of the following correctly declares an array?
A. int anarray[10];
B. int anarray;
C. anarray{10};
D. array anarray[10];

2. What is the index number of the last element of an array with 29 elements?
A. 29
B. 28
C. 0
D. Programmer-defined

3. Which of the following is a two-dimensional array?
A. array anarray[20][20];
B. int anarray[20][20];
C. int array[20, 20];
D. char array[20];

4. Which of the following correctly accesses the seventh element stored in foo, an array with 100 elements?
A. foo[6];
B. foo[7];
C. foo(7);
D. foo;

5. Which of the following gives the memory address of the first element in array foo, an array with 100 elements?
A. foo[0];
B. foo;
C. &foo;
D. foo[1];

1. Which of the following is a string literal?
A. Static String
B. "Static String"
C. 'Static String'
D. char string[100];

2. What character ends all strings?
A. '.'
B. ' '
C. '\0'
D. '\n'

3. Which of the following reads in a string named x with one hundred characters?
A. fgets(x, 101, stdin);
B. fgets(x, 100, stdin);
C. readline(x, 100, '\n');
D. read(x);

4. Which of the following functions compares two strings?
A. compare();
B. stringcompare();
C. cmp();
D. strcmp();

5. Which of the following adds one string to the end of another?
A. append();
B. stringadd();
C. strcat();
D. stradd();

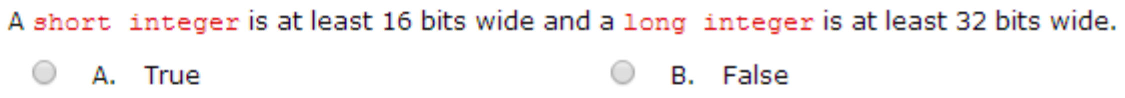
1. What object do you use to represent a file in C?
A. FILE*
B. fopen
C. printf
D. fprintf

2. Before you can read or write to a file in C, what do you need to do?
A. Call fopen on the file
B. Create the file
C. Call fclose on the file
D. Use fprintf

3. How do you write a string of text into a file?
A. Open file and use fprintf.
B. Open a file and use printf, the output will go to the file instead of the screen.
C. Open a file, and use fputc repeatedly.
D. Use fread to read data into the file.

4. What flag do you need to pass to fopen to append to an existing file instead of recreating the file?
A. a
B. r
C. w
D. W+

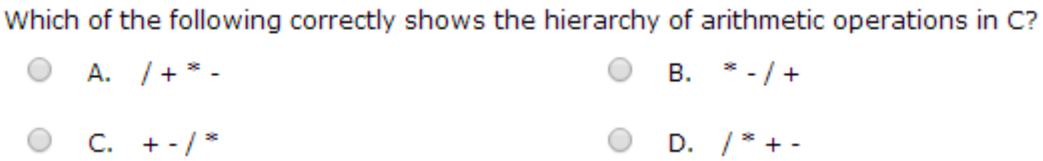
5. How do you open a file for binary IO?
A. Use the "b" flag to fopen
B. Files are opened for binary IO by default
C. Use fread and fwrite
D. You need to use a special file type, BINARYFILE*



A `short integer` is at least 16 bits wide and a `long integer` is at least 32 bits wide.

○ A. True

○ B. False



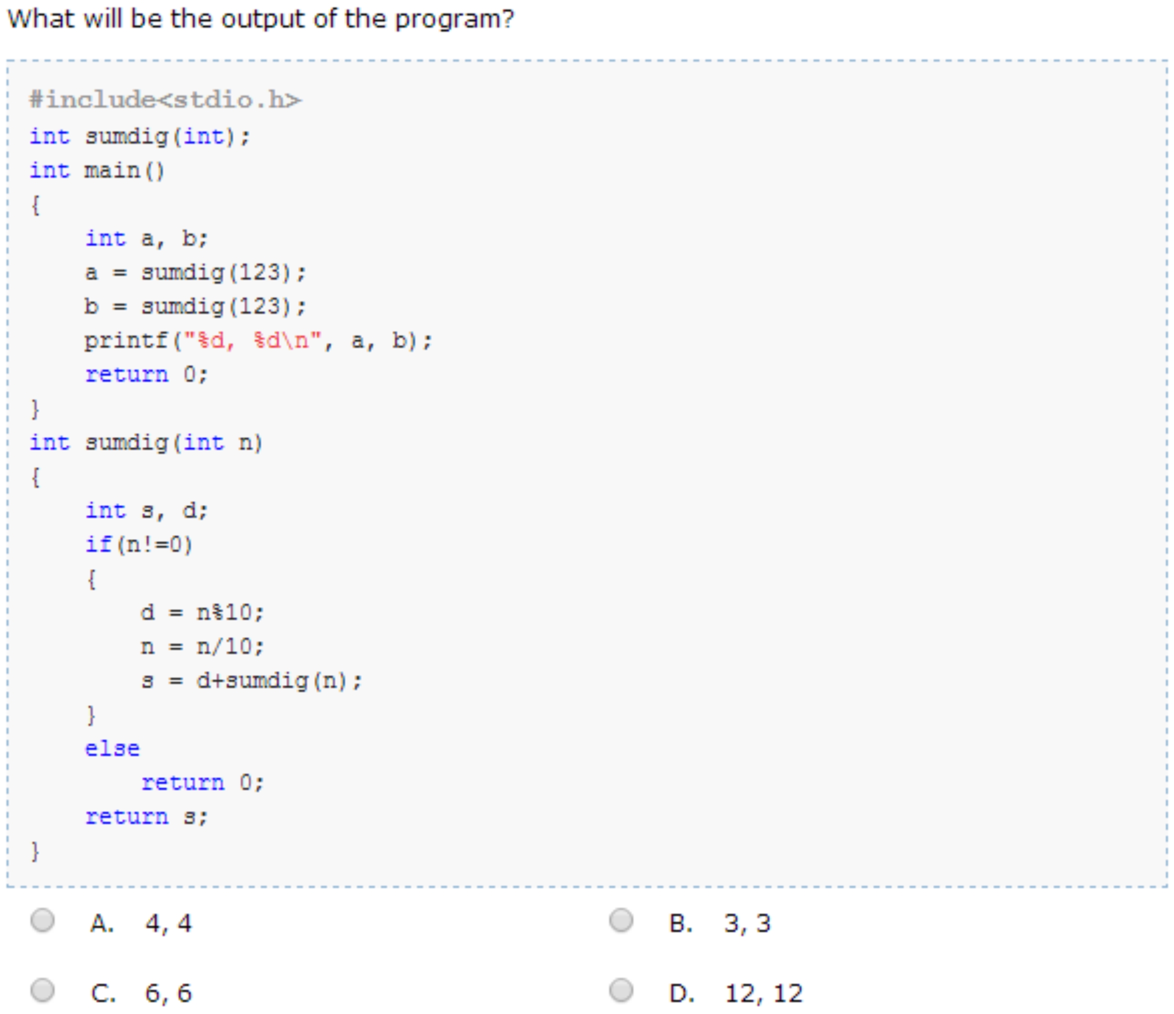
Which of the following correctly shows the hierarchy of arithmetic operations in C?

○ A. / + * -

○ B. * - / +

○ C. + - / *

○ D. / * + -



What will be the output of the program?

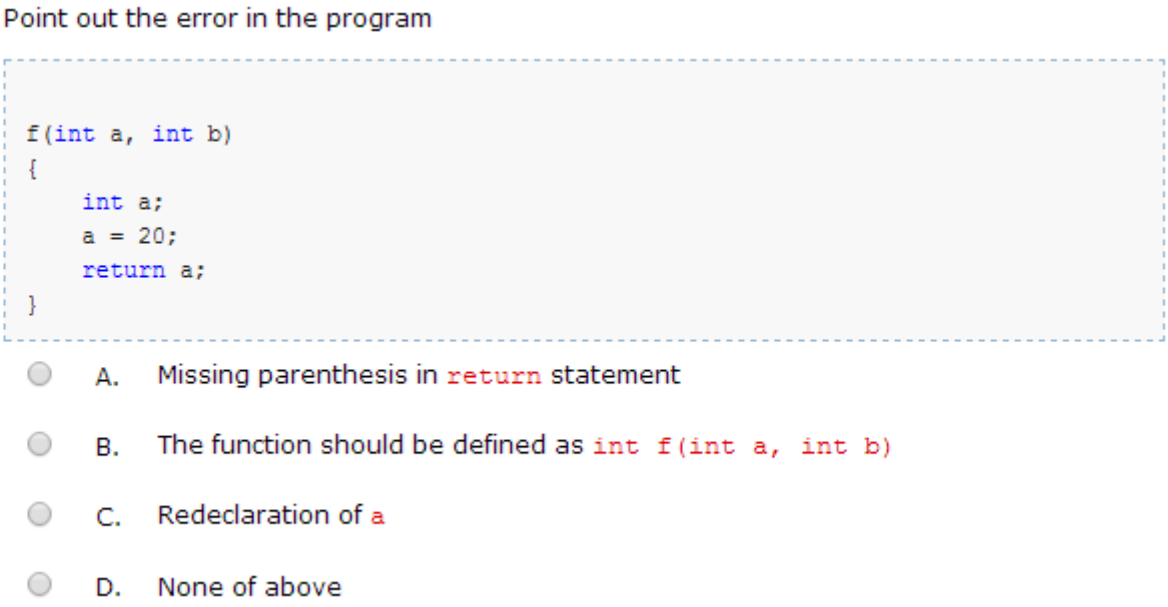
```
#include<stdio.h>
int sumdig(int);
int main()
{
    int a, b;
    a = sumdig(123);
    b = sumdig(123);
    printf("%d, %d\n", a, b);
    return 0;
}
int sumdig(int n)
{
    int s, d;
    if(n!=0)
    {
        d = n%10;
        n = n/10;
        s = d+sumdig(n);
    }
    else
        return 0;
    return s;
}
```

○ A. 4, 4

○ B. 3, 3

○ C. 6, 6

○ D. 12, 12



Point out the error in the program

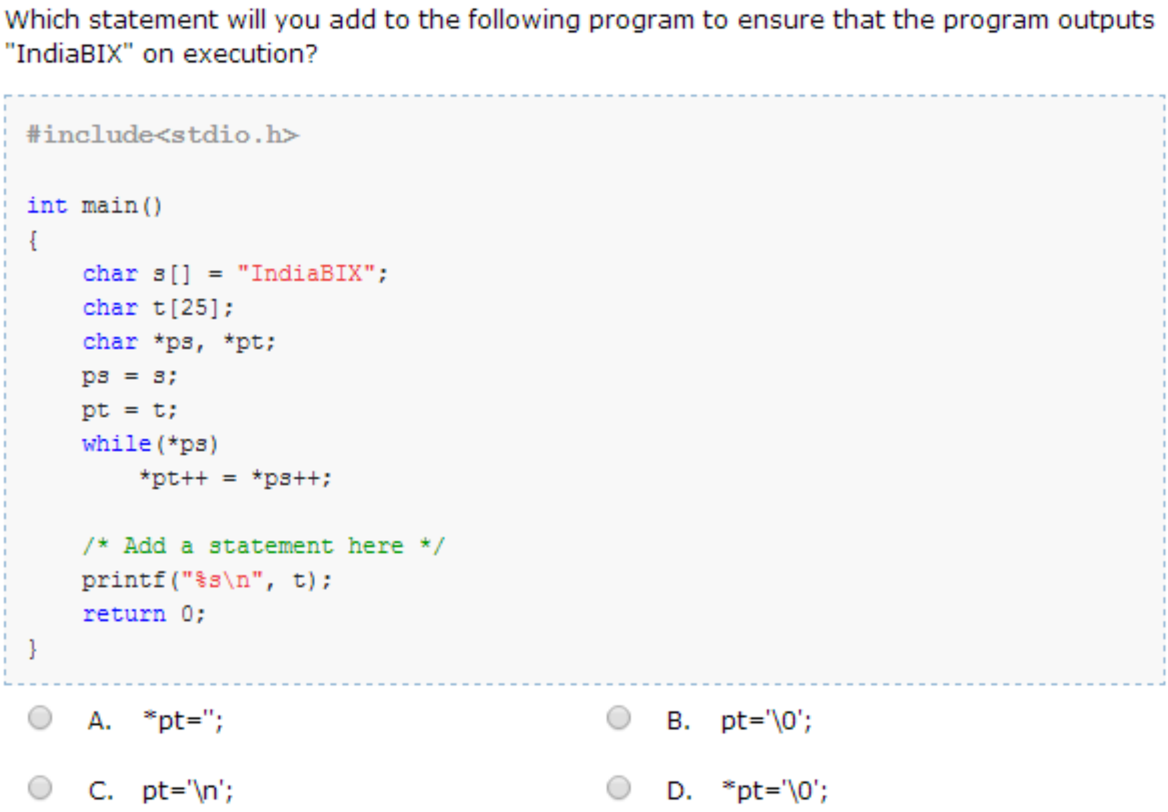
```
f(int a, int b)
{
    int a;
    a = 20;
    return a;
}
```

○ A. Missing parenthesis in `return` statement

○ B. The function should be defined as `int f(int a, int b)`

○ C. Redeclaration of `a`

○ D. None of above



Which statement will you add to the following program to ensure that the program outputs "IndiaBIX" on execution?

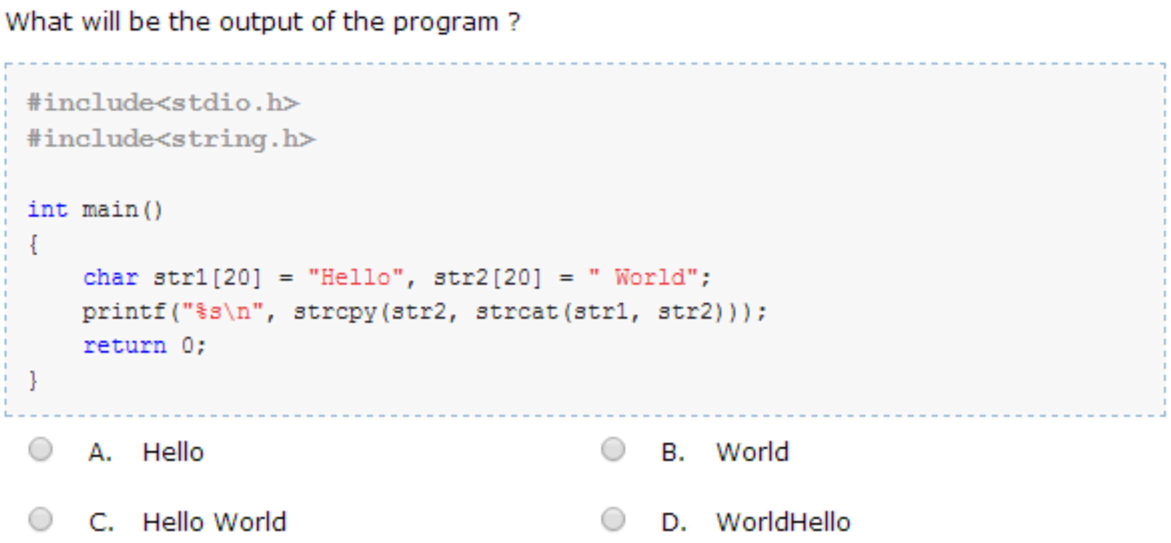
```
#include<stdio.h>

int main()
{
    char s[] = "IndiaBIX";
    char t[25];
    char *ps, *pt;
    ps = s;
    pt = t;
    while(*ps)
        *pt++ = *ps++;

    /* Add a statement here */
    printf("%s\n", t);
    return 0;
}
```

○ A. *pt='';

○ B. pt='\0';

○ C. pt='\n';

○ D. *pt='\0';



What will be the output of the program ?

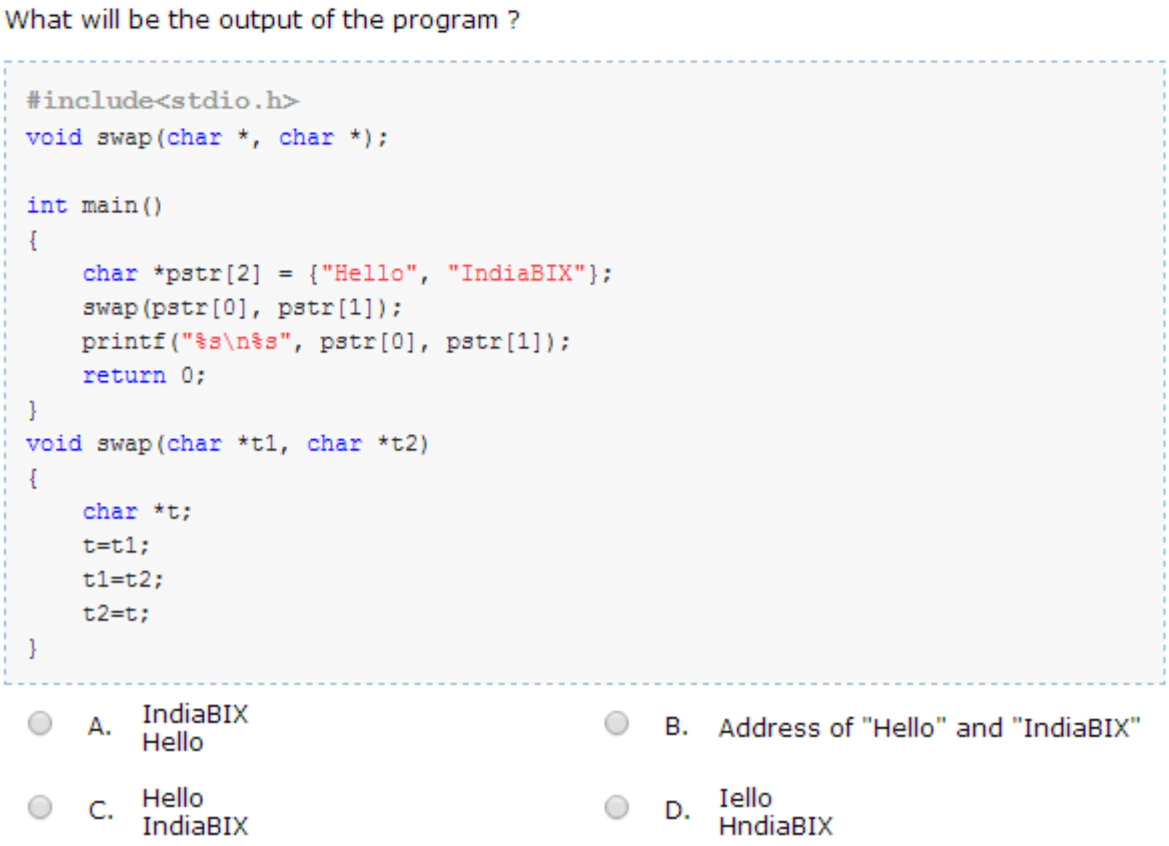
```
#include<stdio.h>
#include<string.h>

int main()
{
    char str1[20] = "Hello", str2[20] = " World";
    printf("%s\n", strcpy(str2, strcat(str1, str2)));
    return 0;
}
```

○ A. Hello

○ B. World

○ C. Hello World

○ D. WorldHello



What will be the output of the program ?

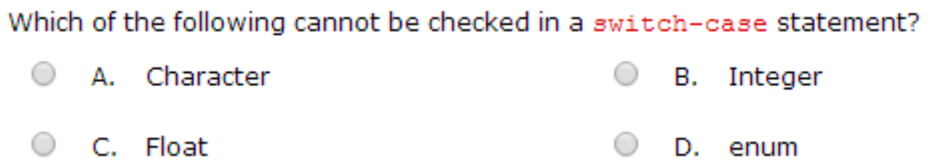
```
#include<stdio.h>
void swap(char *, char *);

int main()
{
    char *pstr[2] = {"Hello", "IndiaBIX"};
    swap(pstr[0], pstr[1]);
    printf("%s\n%s", pstr[0], pstr[1]);
    return 0;
}
void swap(char *t1, char *t2)
{
    char *t;
    t=t1;
    t1=t2;
    t2=t;
}
```

○ A. IndiaBIX
Hello

○ B. Address of "Hello" and "IndiaBIX"

○ C. Hello
IndiaBIX

○ D. Iello
HndiaBIX



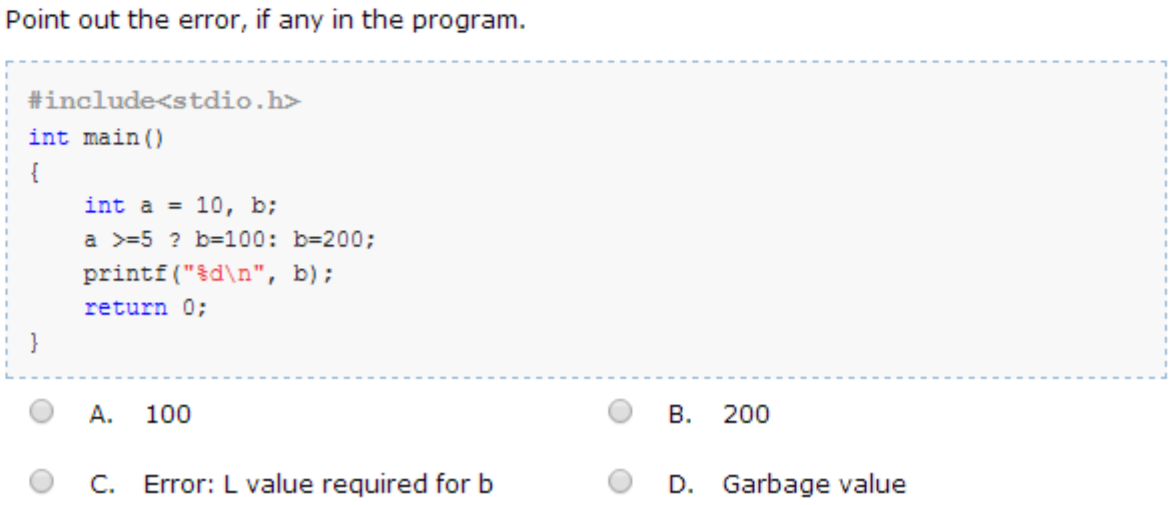
Which of the following cannot be checked in a `switch-case` statement?

○ A. Character

○ B. Integer

○ C. Float

○ D. enum

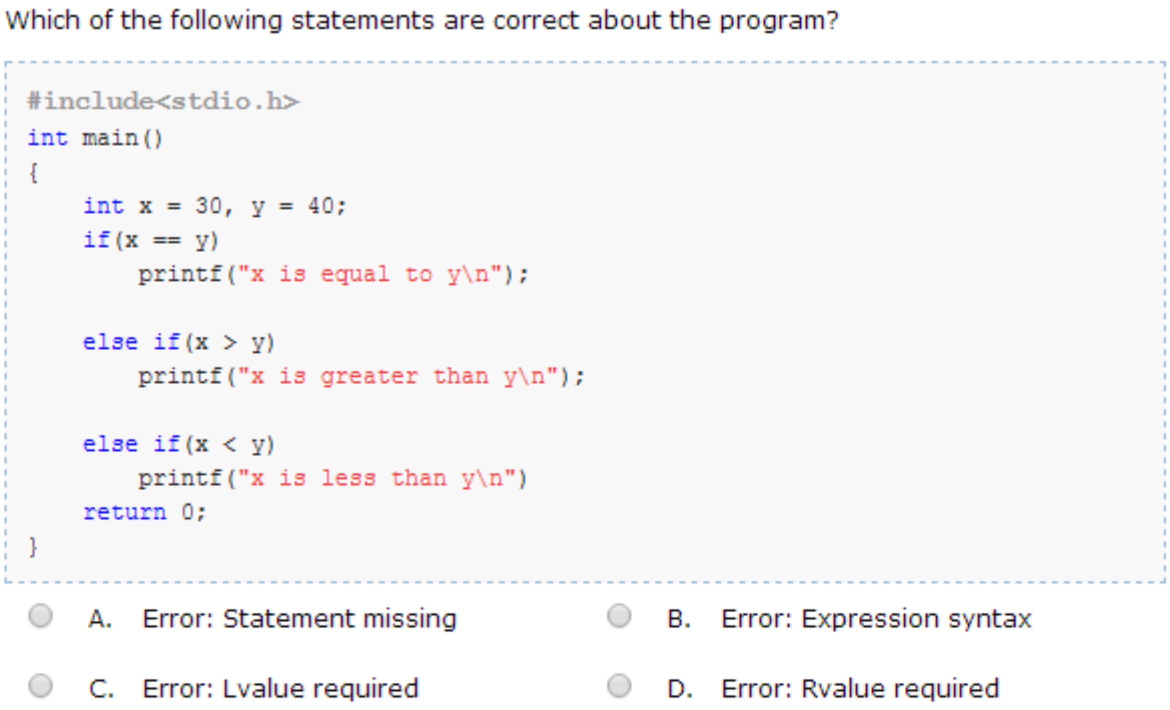


Point out the error, if any in the program.

```
#include<stdio.h>
int main()
{
    int a = 10, b;
    a >=5 ? b=100: b=200;
    printf("%d\n", b);
    return 0;
}
```

○ A. 100 ○ B. 200

○ C. Error: L value required for b ○ D. Garbage value

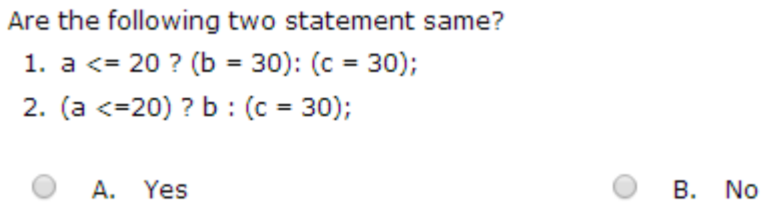


Which of the following statements are correct about the program?

```
#include<stdio.h>
int main()
{
    int x = 30, y = 40;
    if(x == y)
        printf("x is equal to y\n");

    else if(x > y)
        printf("x is greater than y\n");

    else if(x < y)
        printf("x is less than y\n")
    return 0;
}
```

○ A. Error: Statement missing ○ B. Error: Expression syntax

○ C. Error: Lvalue required ○ D. Error: Rvalue required

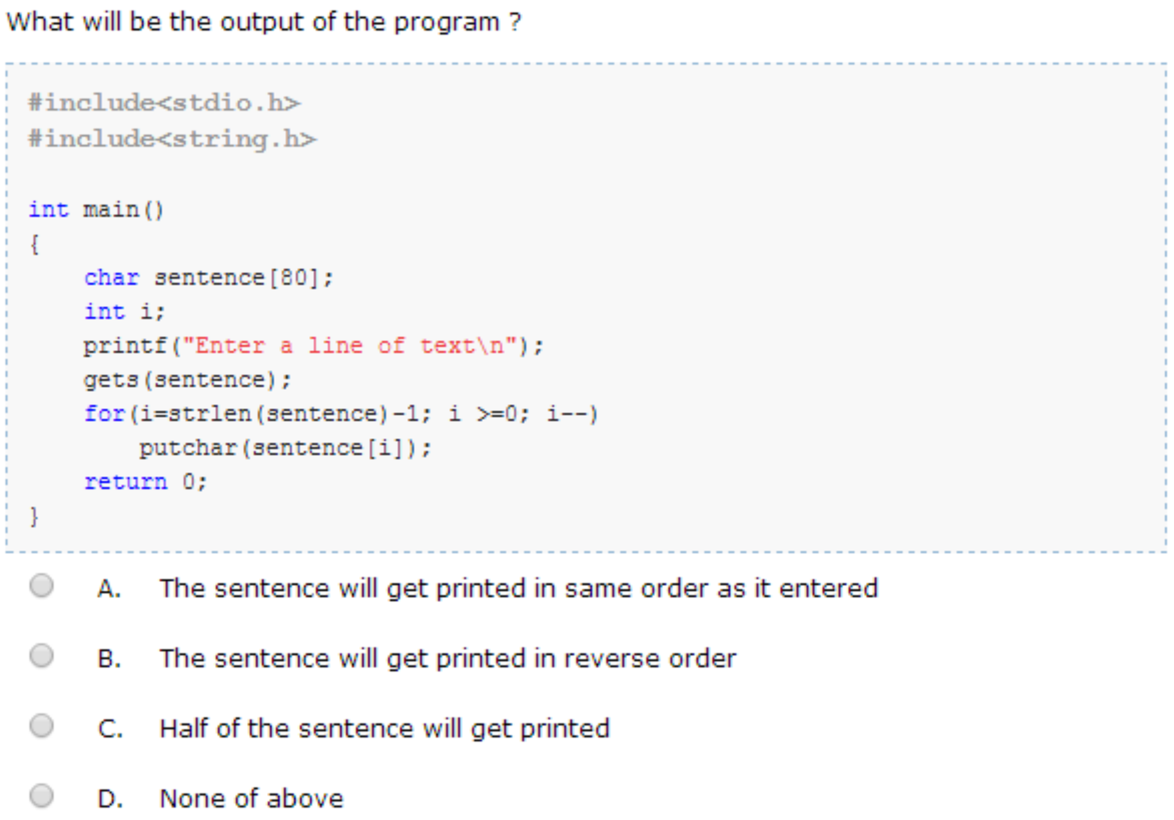


Are the following two statement same?

1. a <= 20 ? (b = 30): (c = 30);
2. (a <=20) ? b : (c = 30);

○ A. Yes

○ B. No



What will be the output of the program ?

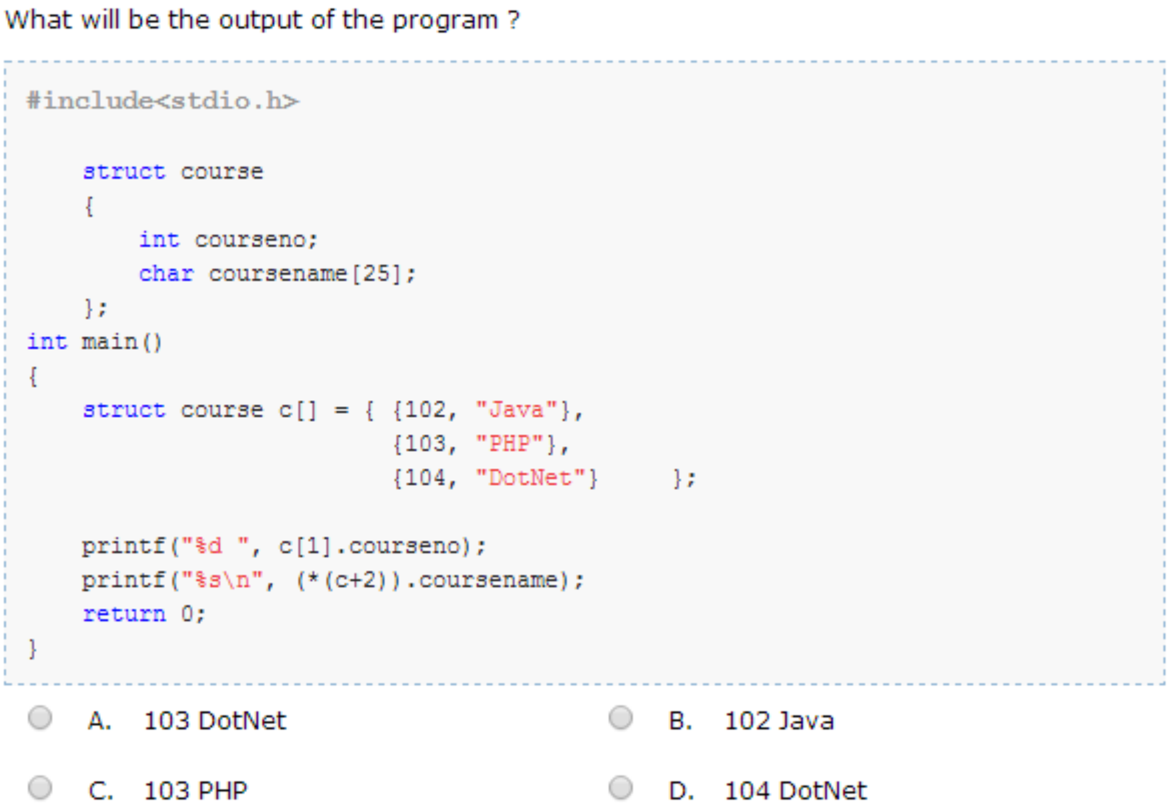
```
#include<stdio.h>
#include<string.h>

int main()
{
    char sentence[80];
    int i;
    printf("Enter a line of text\n");
    gets(sentence);
    for(i=strlen(sentence)-1; i >=0; i--)
        putchar(sentence[i]);
    return 0;
}
```

○ A. The sentence will get printed in same order as it entered

○ B. The sentence will get printed in reverse order

○ C. Half of the sentence will get printed

○ D. None of above



What will be the output of the program ?

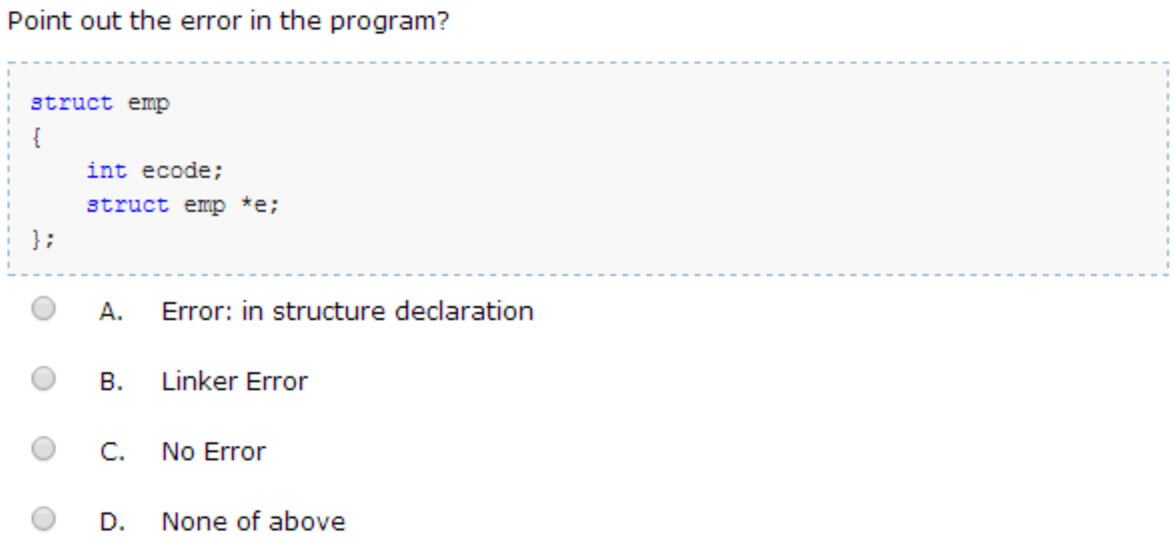
```
#include<stdio.h>

    struct course
    {
        int courseno;
        char coursename[25];
    };
int main()
{
    struct course c[] = { {102, "Java"},
                          {103, "PHP"},
                          {104, "DotNet"}      };

    printf("%d ", c[1].courseno);
    printf("%s\n", (*(c+2)).coursename);
    return 0;
}
```

○ A. 103 DotNet

○ B. 102 Java

○ C. 103 PHP

○ D. 104 DotNet



Point out the error in the program?

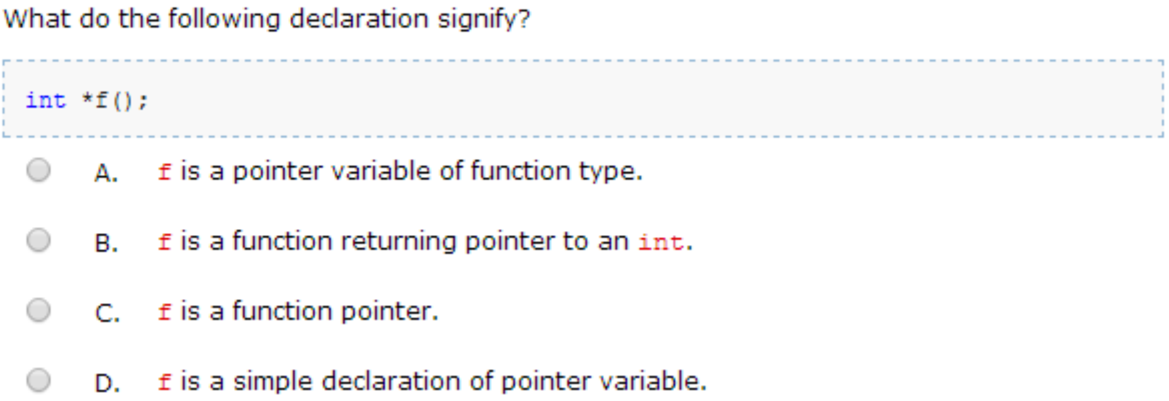
```
struct emp
{
    int ecode;
    struct emp *e;
};
```

○ A. Error: in structure declaration

○ B. Linker Error

○ C. No Error

○ D. None of above



What do the following declaration signify?

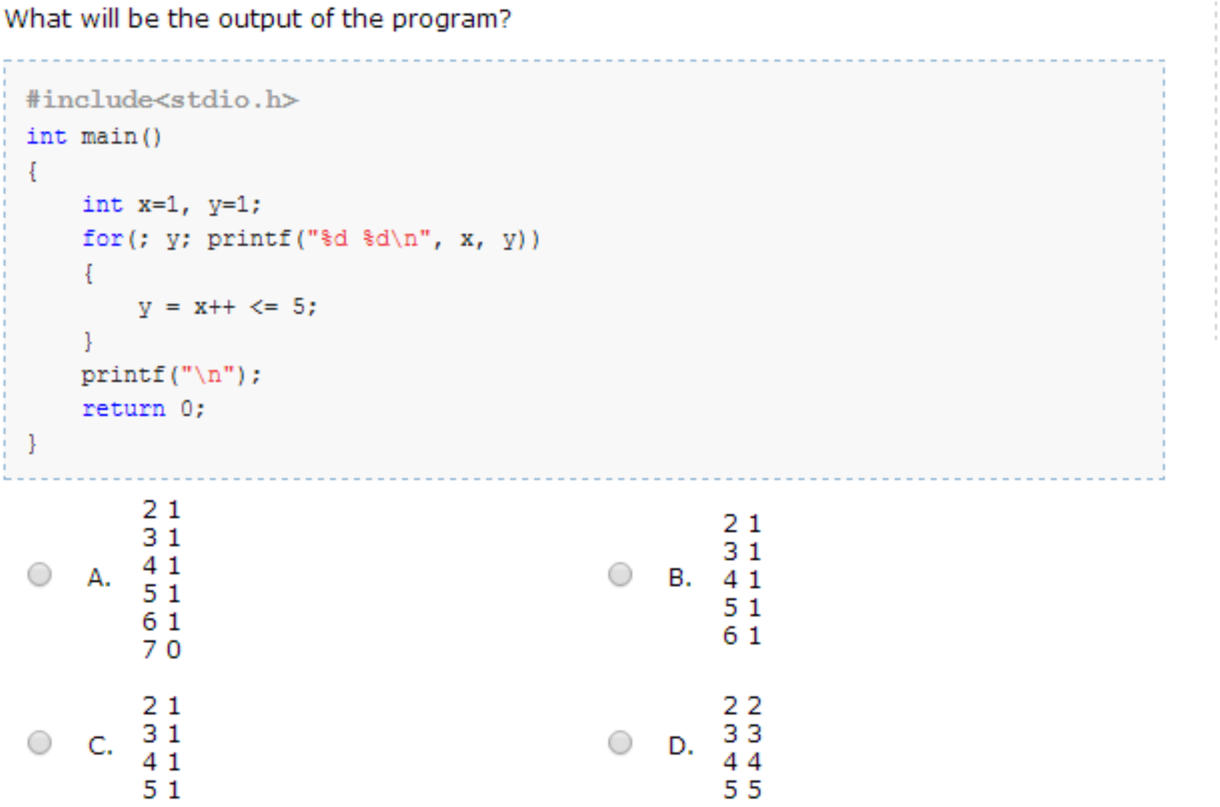
```
int *f();
```

○ A. f is a pointer variable of function type.

○ B. f is a function returning pointer to an int.

○ C. f is a function pointer.

○ D. f is a simple declaration of pointer variable.



What will be the output of the program?

```
#include<stdio.h>
int main()
{
    int x=1, y=1;
    for(; y; printf("%d %d\n", x, y))
    {
        y = x++ <= 5;
    }
    printf("\n");
    return 0;
}
```

○ A.

```
2 1
3 1
4 1
5 1
6 1
7 0
```

○ B.

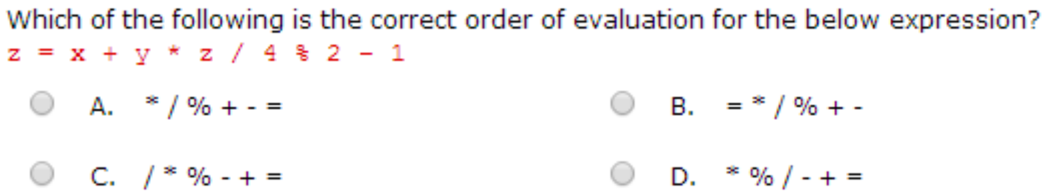
```
2 1
3 1
4 1
5 1
6 1
```

○ C.

```
2 1
3 1
4 1
5 1
```

○ D.

```
2 2
3 3
4 4
5 5
```



Which of the following is the correct order of evaluation for the below expression?

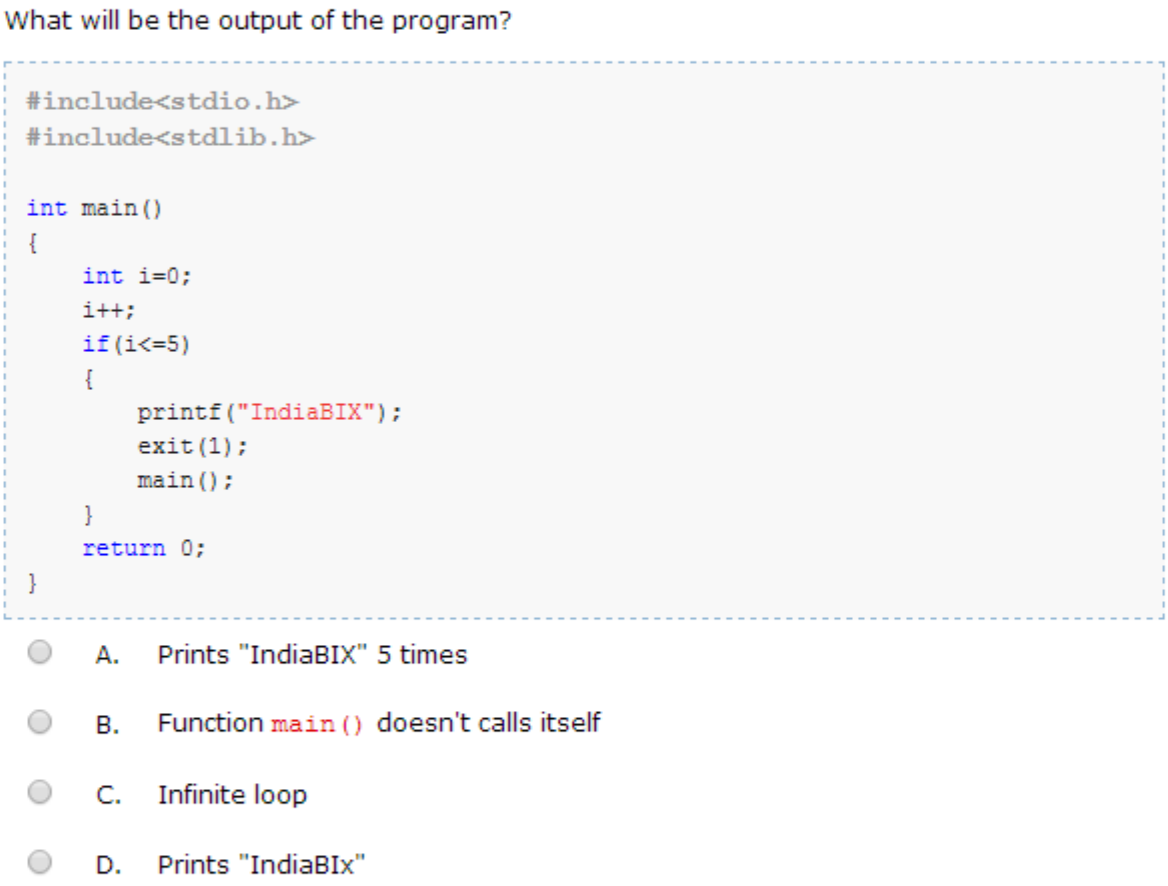
```
z = x + y * z / 4 % 2 - 1
```

○ A. * / % + - =

○ B. = * / % + -

○ C. / * % - + =

○ D. * % / - + =



What will be the output of the program?

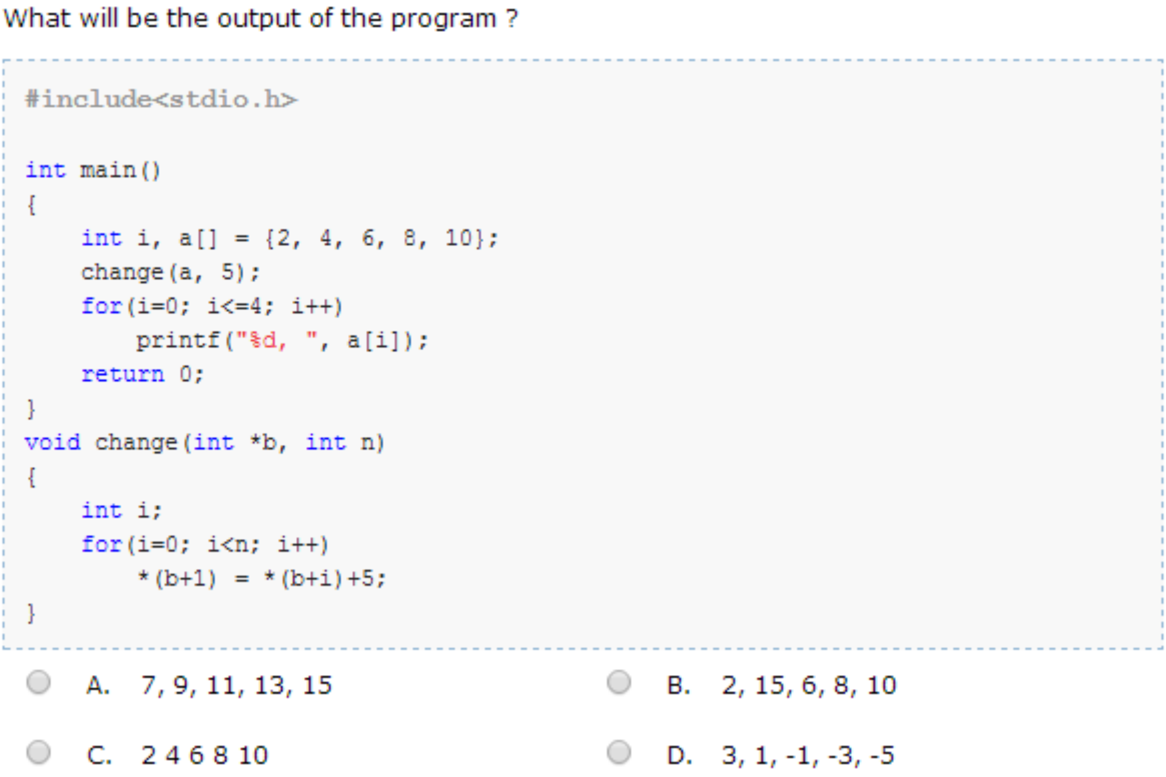
```
#include<stdio.h>
#include<stdlib.h>

int main()
{
    int i=0;
    i++;
    if(i<=5)
    {
        printf("IndiaBIX");
        exit(1);
        main();
    }
    return 0;
}
```

○ A. Prints "IndiaBIX" 5 times

○ B. Function `main()` doesn't calls itself

○ C. Infinite loop

○ D. Prints "IndiaBIx"

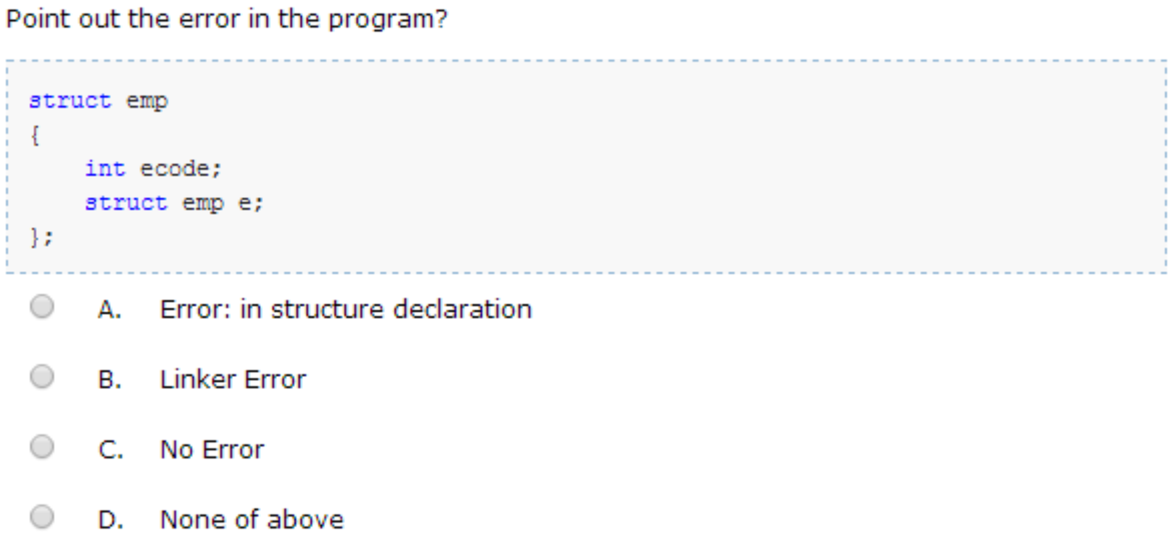


What will be the output of the program ?

```
#include<stdio.h>

int main()
{
    int i, a[] = {2, 4, 6, 8, 10};
    change(a, 5);
    for(i=0; i<=4; i++)
        printf("%d, ", a[i]);
    return 0;
}
void change(int *b, int n)
{
    int i;
    for(i=0; i<n; i++)
        *(b+1) = *(b+i)+5;
}
```

○ A. 7, 9, 11, 13, 15 ○ B. 2, 15, 6, 8, 10

○ C. 2 4 6 8 10 ○ D. 3, 1, -1, -3, -5



Point out the error in the program?

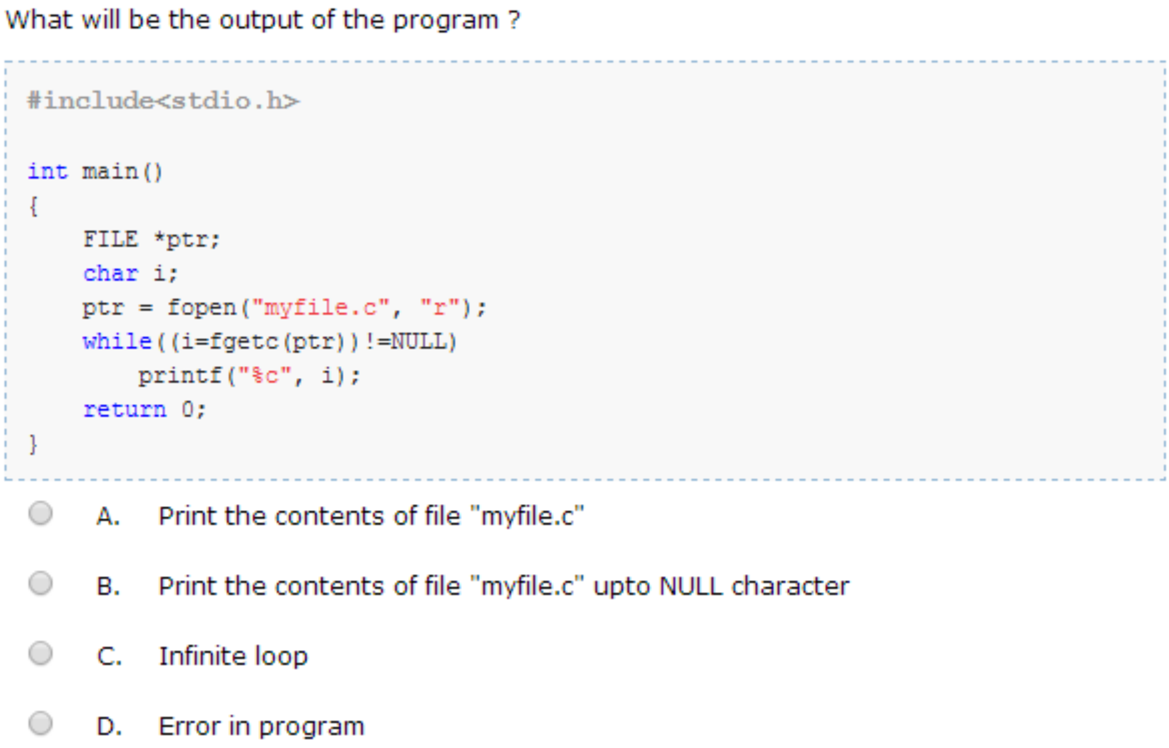
```
struct emp
{
    int ecode;
    struct emp e;
};
```

○ A. Error: in structure declaration

○ B. Linker Error

○ C. No Error

○ D. None of above



What will be the output of the program ?

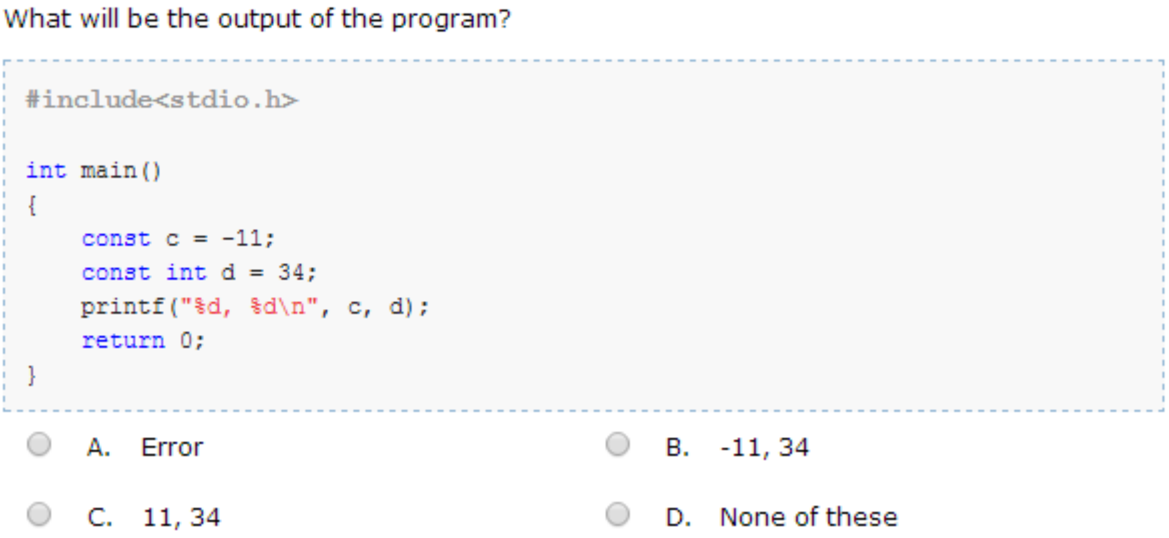
```
#include<stdio.h>

int main()
{
    FILE *ptr;
    char i;
    ptr = fopen("myfile.c", "r");
    while((i=fgetc(ptr))!=NULL)
        printf("%c", i);
    return 0;
}
```

○ A. Print the contents of file "myfile.c"

○ B. Print the contents of file "myfile.c" upto NULL character

○ C. Infinite loop

○ D. Error in program



What will be the output of the program?

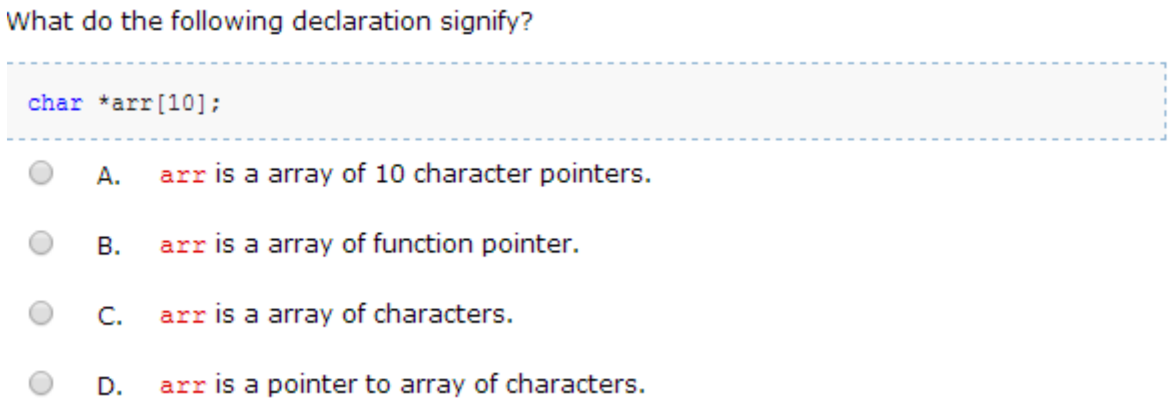
```
#include<stdio.h>

int main()
{
    const c = -11;
    const int d = 34;
    printf("%d, %d\n", c, d);
    return 0;
}
```

○ A. Error

○ B. -11, 34

○ C. 11, 34

○ D. None of these



What do the following declaration signify?

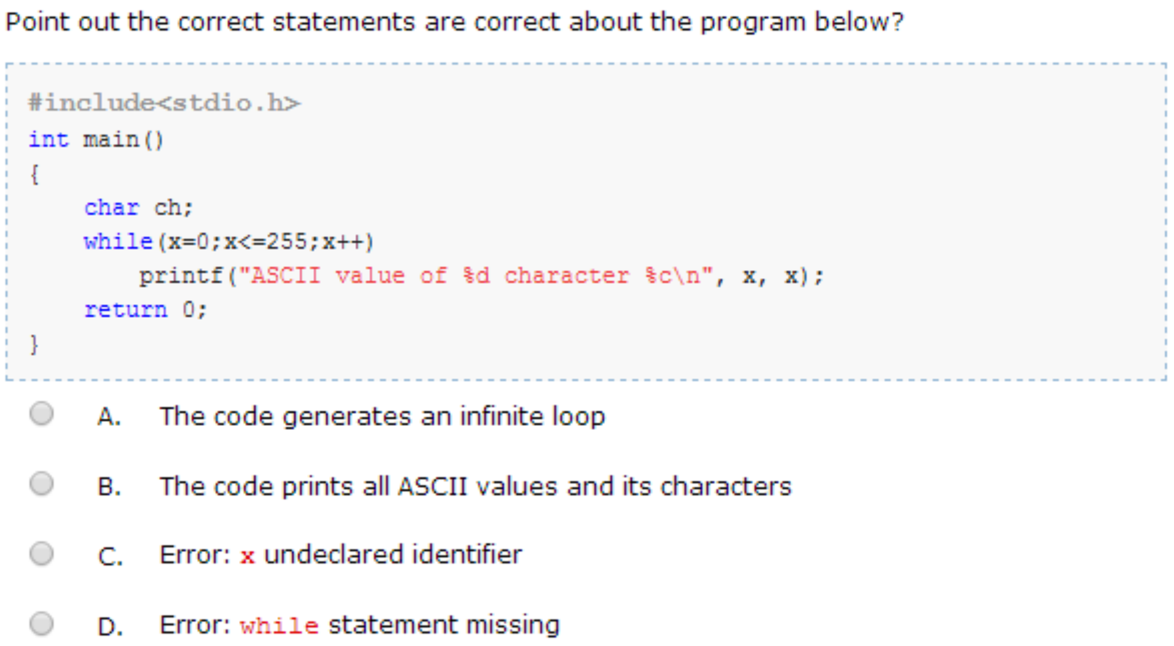
```
char *arr[10];
```

○ A. arr is a array of 10 character pointers.

○ B. arr is a array of function pointer.

○ C. arr is a array of characters.

○ D. arr is a pointer to array of characters.



Point out the correct statements are correct about the program below?

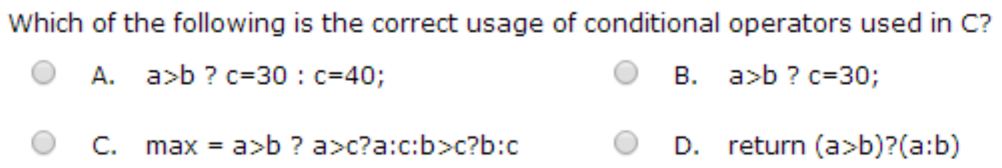
```
#include<stdio.h>
int main()
{
    char ch;
    while(x=0;x<=255;x++)
        printf("ASCII value of %d character %c\n", x, x);
    return 0;
}
```

○ A. The code generates an infinite loop

○ B. The code prints all ASCII values and its characters

○ C. Error: x undeclared identifier

○ D. Error: while statement missing



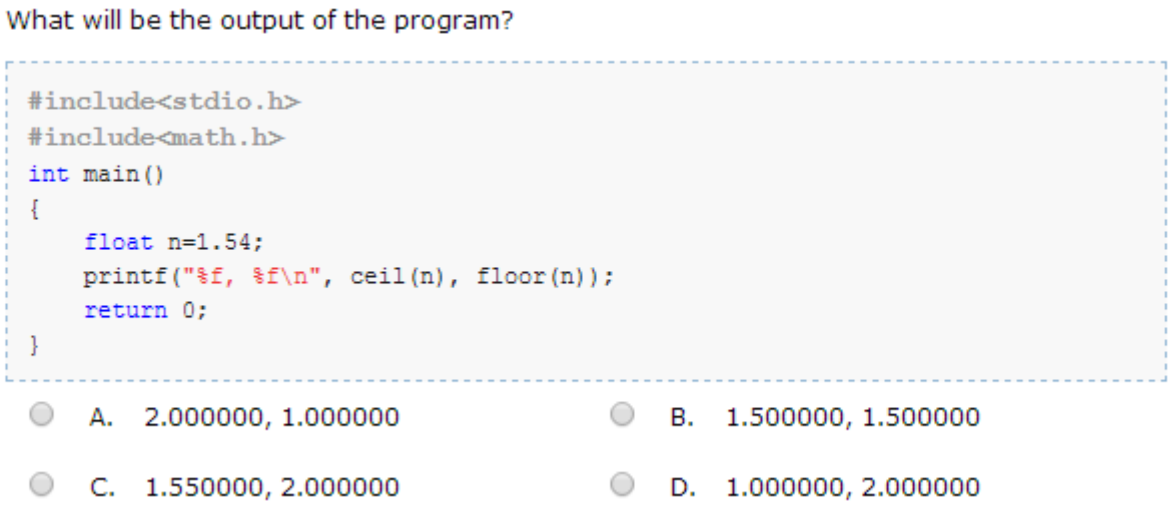
Which of the following is the correct usage of conditional operators used in C?

○ A. a>b ? c=30 : c=40;

○ B. a>b ? c=30;

○ C. max = a>b ? a>c?a:c:b>c?b:c

○ D. return (a>b)?(a:b)



What will be the output of the program?

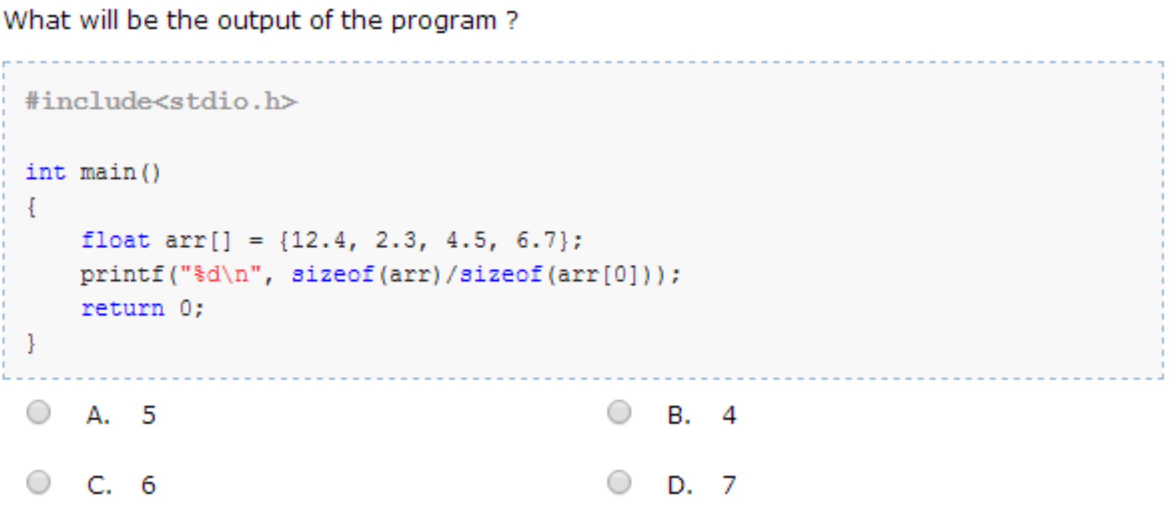
```
#include<stdio.h>
#include<math.h>
int main()
{
    float n=1.54;
    printf("%f, %f\n", ceil(n), floor(n));
    return 0;
}
```

○ A. 2.000000, 1.000000

○ B. 1.500000, 1.500000

○ C. 1.550000, 2.000000

○ D. 1.000000, 2.000000



What will be the output of the program ?

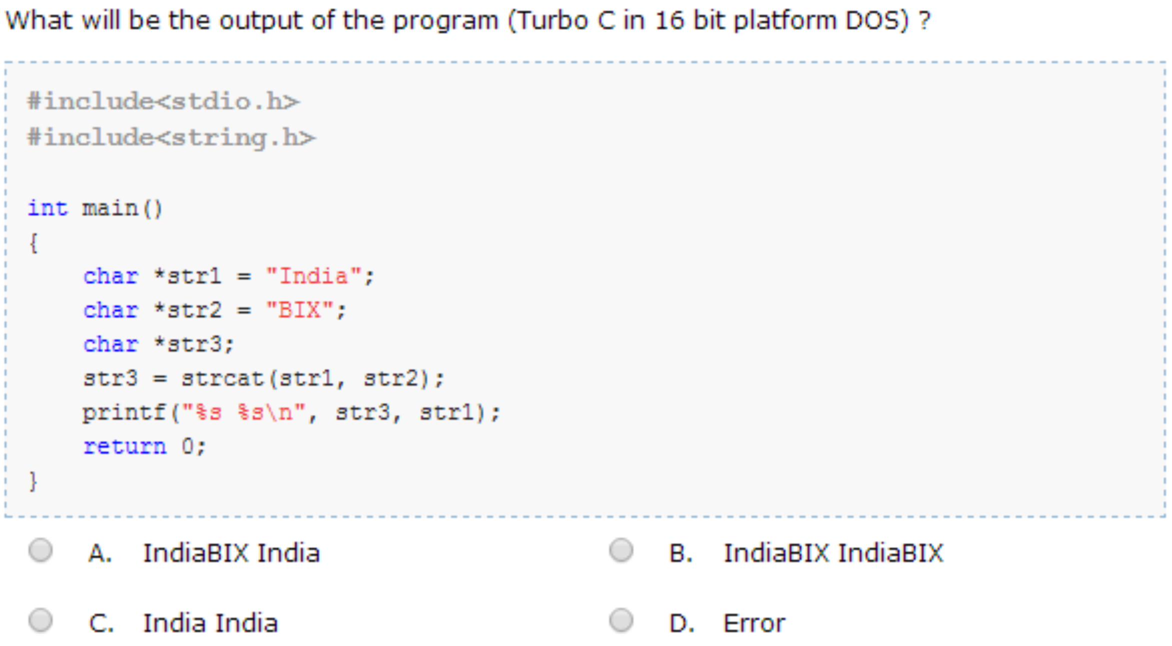
```
#include<stdio.h>

int main()
{
    float arr[] = {12.4, 2.3, 4.5, 6.7};
    printf("%d\n", sizeof(arr)/sizeof(arr[0]));
    return 0;
}
```

○ A. 5

○ B. 4

○ C. 6

○ D. 7



What will be the output of the program (Turbo C in 16 bit platform DOS) ?

```
#include<stdio.h>
#include<string.h>

int main()
{
    char *str1 = "India";
    char *str2 = "BIX";
    char *str3;
    str3 = strcat(str1, str2);
    printf("%s %s\n", str3, str1);
    return 0;
}
```

○ A. IndiaBIX India ○ B. IndiaBIX IndiaBIX

○ C. India India ○ D. Error