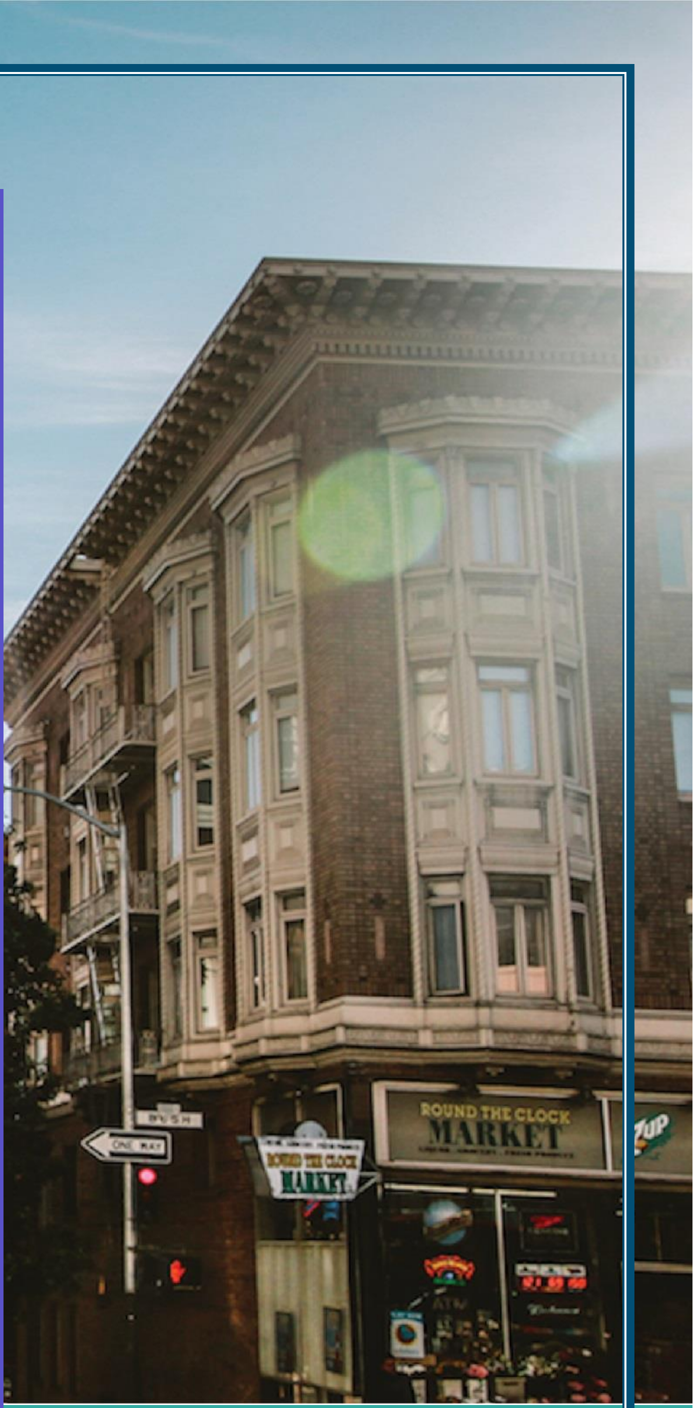
Pointer Challenge

2019

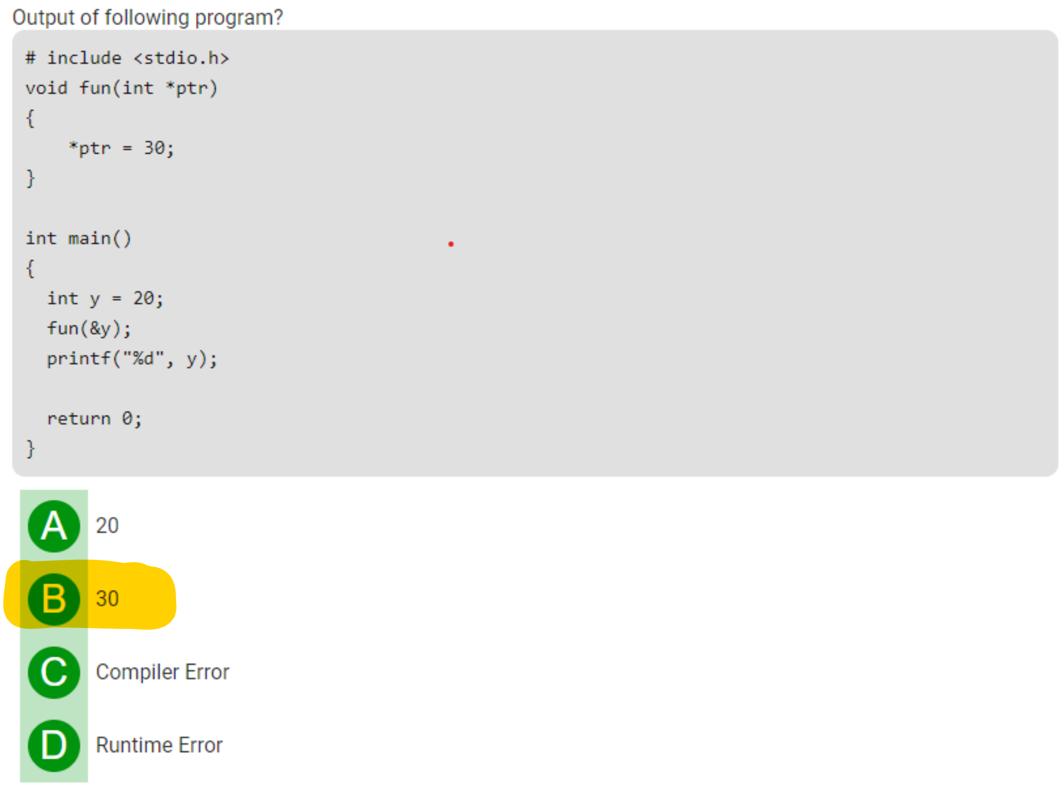


DECEMBER 2

Authored by: Mohamed Adel

DUNGA

1- Choose The Correct Answer :



Output of following program?

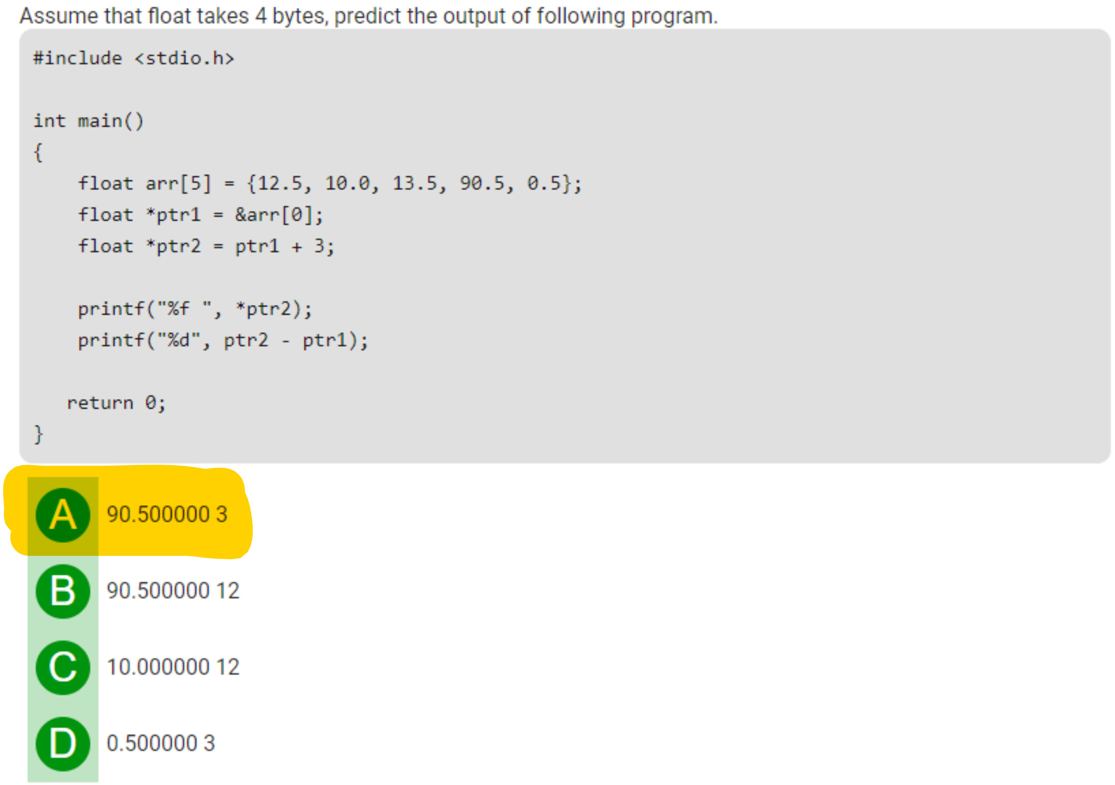
```
# include <stdio.h>
void fun(int *ptr)
{
    *ptr = 30;
}

int main()
{
  int y = 20;
  fun(&y);
  printf("%d", y);

  return 0;
}
```

A 20

[✓ B 30]

C Compiler Error

D Runtime Error



Assume that float takes 4 bytes, predict the output of following program.

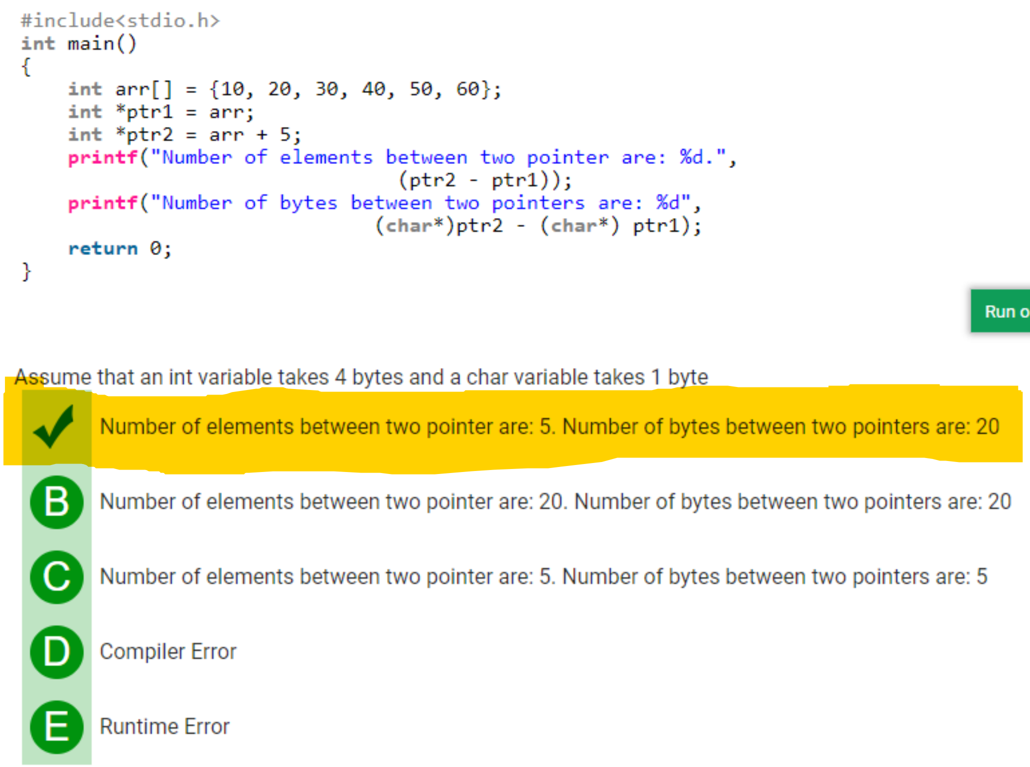
```
#include <stdio.h>

int main()
{
    float arr[5] = {12.5, 10.0, 13.5, 90.5, 0.5};
    float *ptr1 = &arr[0];
    float *ptr2 = ptr1 + 3;

    printf("%f ", *ptr2);
    printf("%d", ptr2 - ptr1);

   return 0;
}
```

[✓ A 90.500000 3]

B 90.500000 12

C 10.000000 12

D 0.500000 3



```
#include<stdio.h>
int main()
{
    int arr[] = {10, 20, 30, 40, 50, 60};
    int *ptr1 = arr;
    int *ptr2 = arr + 5;
    printf("Number of elements between two pointer are: %d.",
                            (ptr2 - ptr1));
    printf("Number of bytes between two pointers are: %d",
                          (char*)ptr2 - (char*) ptr1);
    return 0;
}
```

Assume that an int variable takes 4 bytes and a char variable takes 1 byte

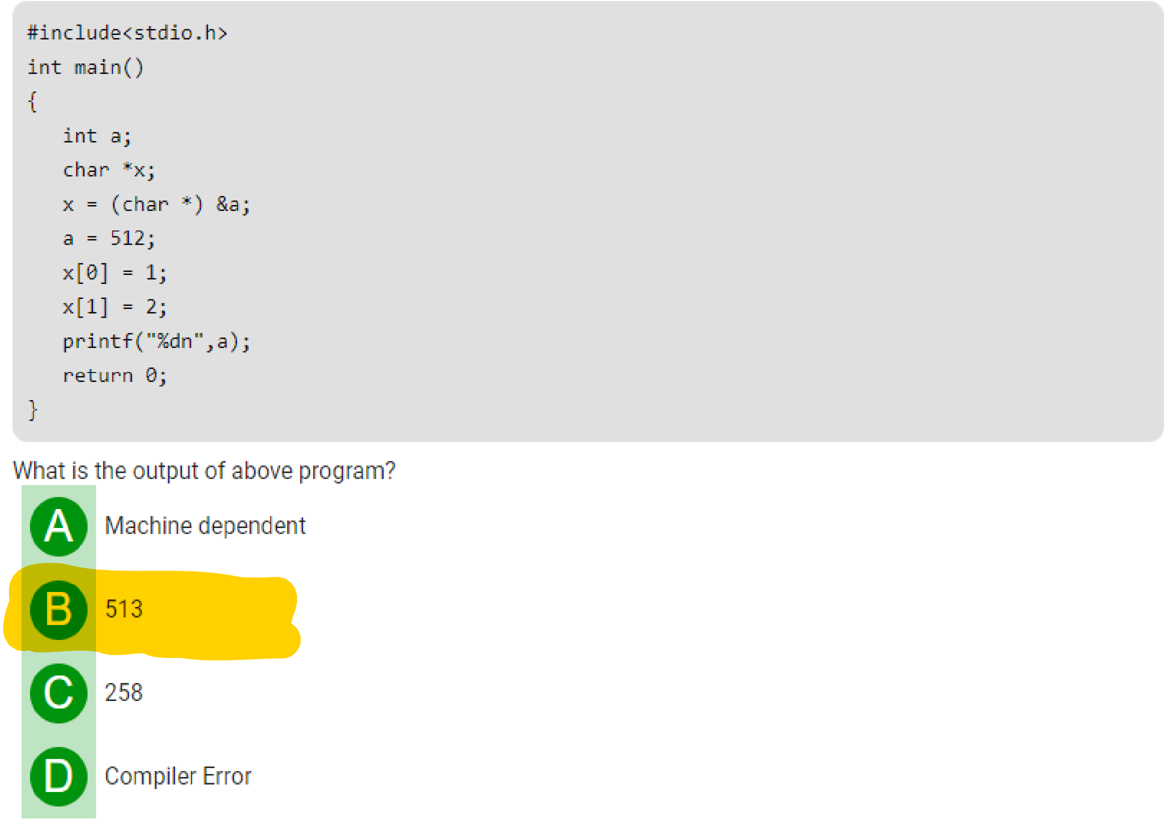
[✓ Number of elements between two pointer are: 5. Number of bytes between two pointers are: 20]

B Number of elements between two pointer are: 20. Number of bytes between two pointers are: 20

C Number of elements between two pointer are: 5. Number of bytes between two pointers are: 5

D Compiler Error

E Runtime Error



```
#include<stdio.h>
int main()
{
   int a;
   char *x;
   x = (char *) &a;
   a = 512;
   x[0] = 1;
   x[1] = 2;
   printf("%dn",a);
   return 0;
}
```

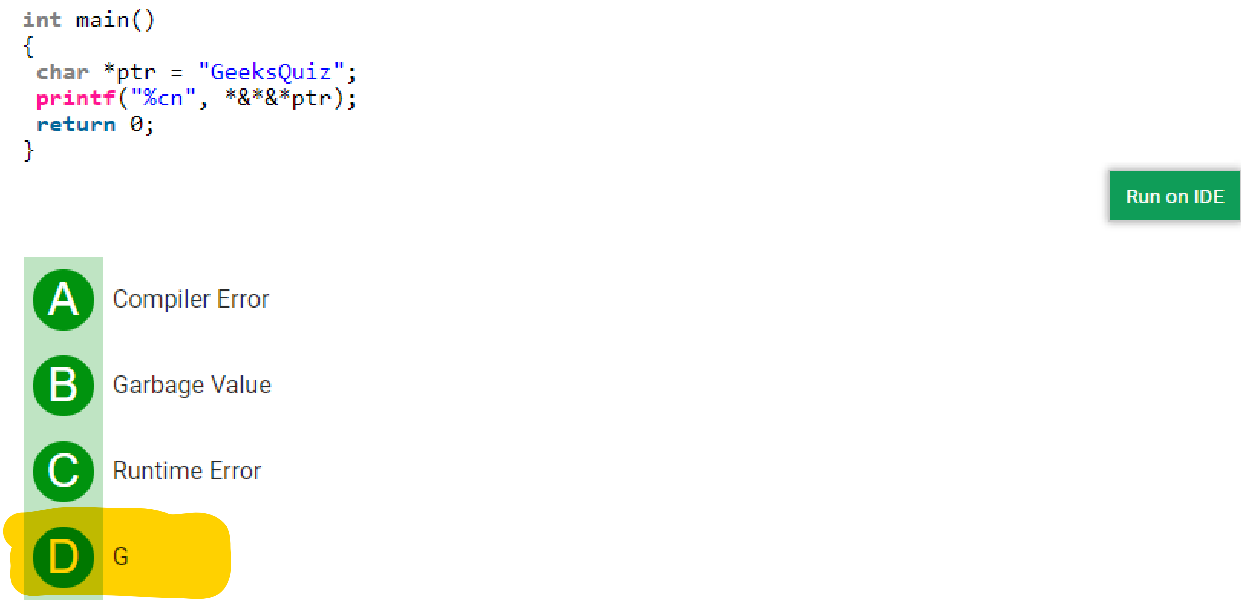
What is the output of above program?

A Machine dependent

[✓ B 513]

C 258

D Compiler Error



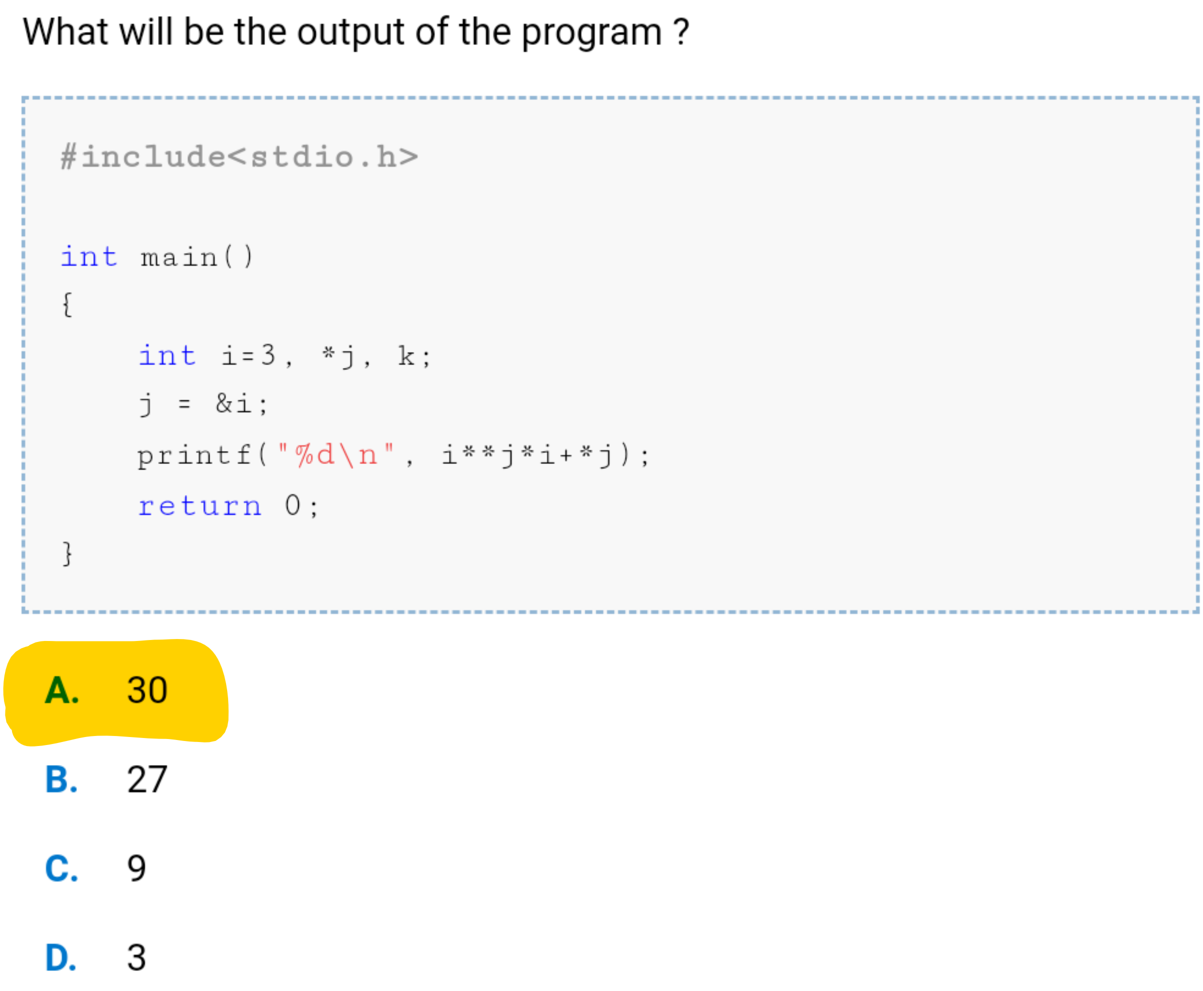
```
int main()
{
 char *ptr = "GeeksQuiz";
 printf("%cn", *&*&*ptr);
 return 0;
}
```

A Compiler Error

B Garbage Value

C Runtime Error

[✓ D G]



What will be the output of the program ?

```
#include<stdio.h>

int main()
{
    int i=3, *j, k;
    j = &i;
    printf("%d\n", i**j*i+*j);
    return 0;
}
```

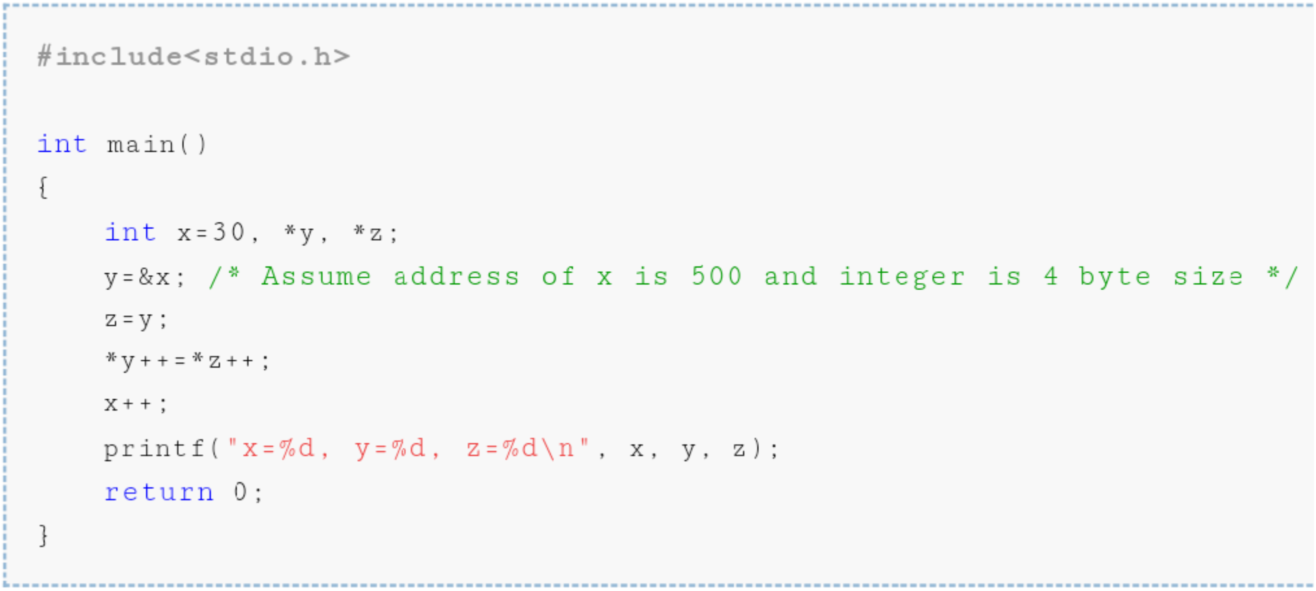
[✓ A. 30]

B. 27

C. 9

D. 3

. 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;
}
```

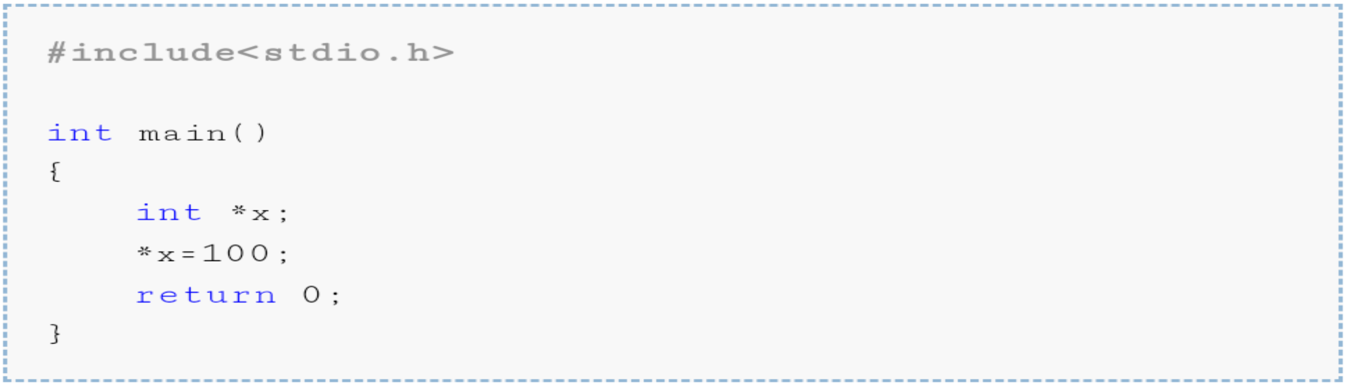
A. x=31, y=502, z=502

B. x=31, y=500, z=500

C. x=31, y=498, z=498

[✓ D. x=31, y=504, z=504]

Point out the compile time error in the program given below.



```
#include<stdio.h>

int main()
{
    int *x;
    *x=100;
    return 0;
}
```

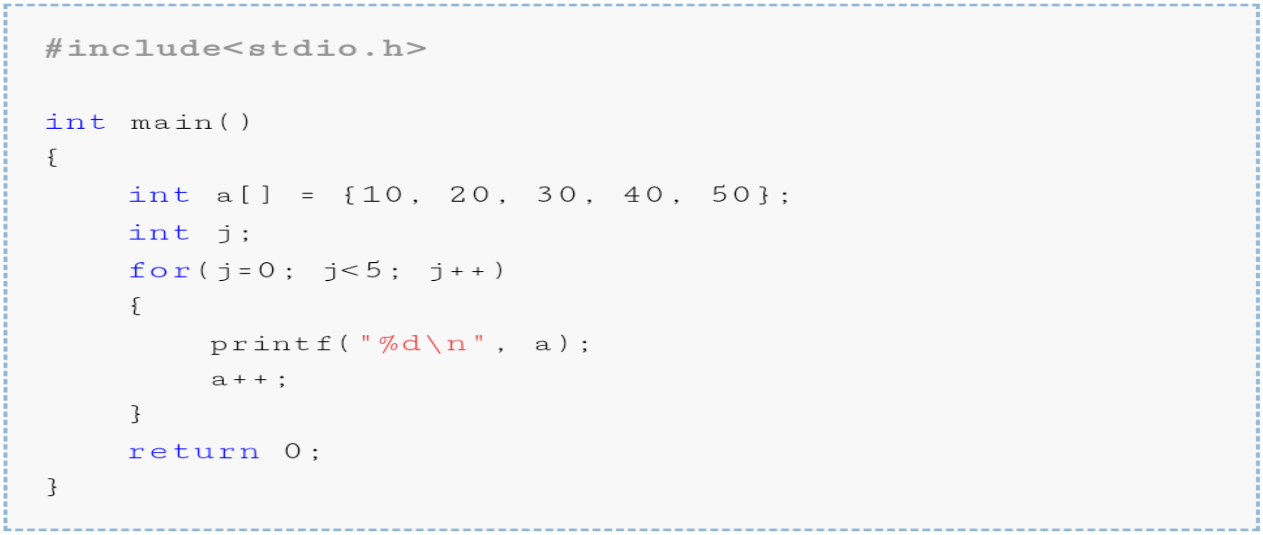
A. Error: invalid assignment for x

B. Error: suspicious pointer conversion

[✓ C. No error]

D. None of above

Point out the error in the program



```
#include<stdio.h>

int main()
{
    int a[] = {10, 20, 30, 40, 50};
    int j;
    for(j=0; j<5; j++)
    {
        printf("%d\n", a);
        a++;
    }
    return 0;
}
```

A. Error: Declaration syntax

B. Error: Expression syntax

C. Error: LValue required

D. Error: Rvalue required

Is there any difference between the following two statements?

```
char *p=0;
char *t=NULL;
```

A. Yes

B. No

Is this a correct way for NULL pointer assignment?

```
int i=0;
char *q=(char*)i;
```

A. Yes

B. No

What is (void*)0?

A. Representation of NULL pointer

B. Representation of void pointer

C. Error

D. None of above

In which header file is the NULL macro defined?

A. stdio.h

B. stddef.h

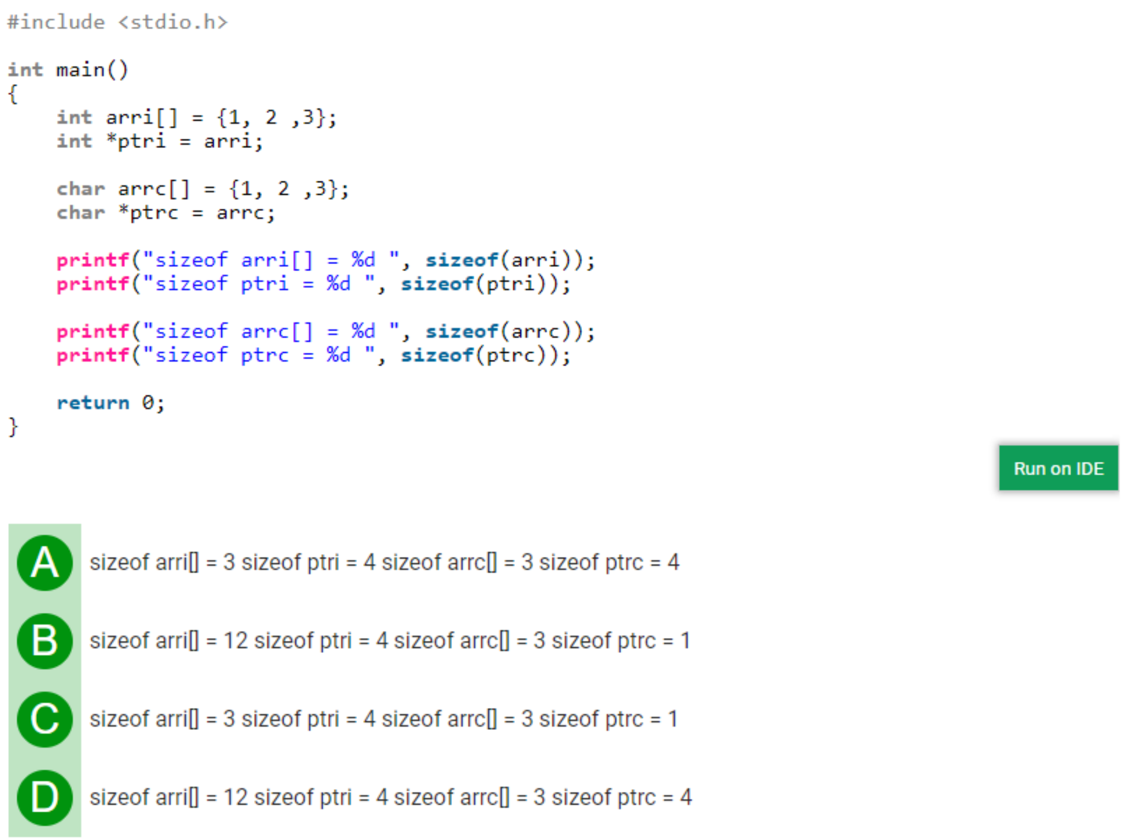
C. stdio.h and stddef.h

D. math.h

Is the NULL pointer same as an uninitialised pointer?

A. Yes

B. No



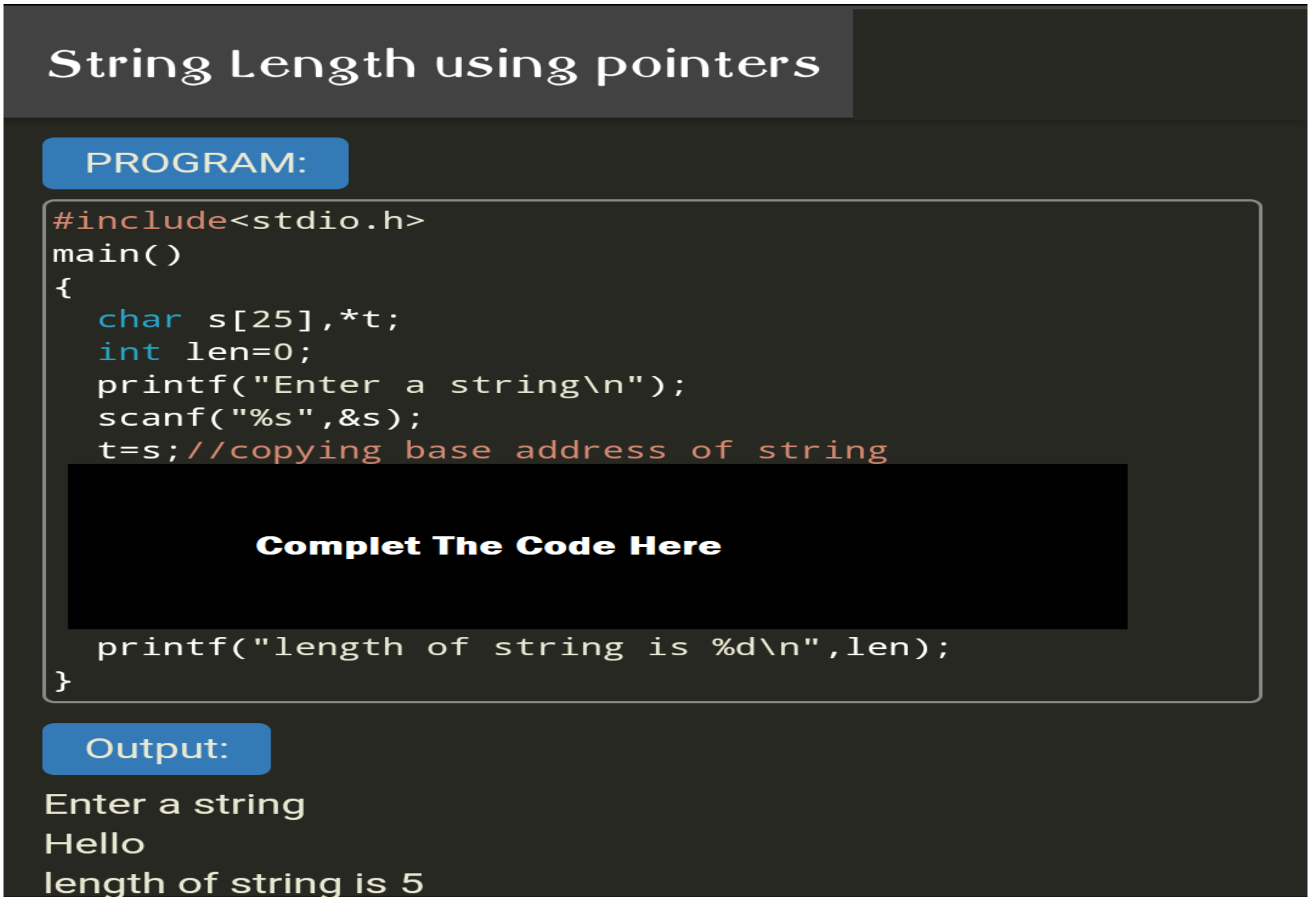
```
#include <stdio.h>

int main()
{
    int arri[] = {1, 2 ,3};
    int *ptri = arri;

    char arrc[] = {1, 2 ,3};
    char *ptrc = arrc;

    printf("sizeof arri[] = %d ", sizeof(arri));
    printf("sizeof ptri = %d ", sizeof(ptri));

    printf("sizeof arrc[] = %d ", sizeof(arrc));
    printf("sizeof ptrc = %d ", sizeof(ptrc));

    return 0;
}
```

A sizeof arri[] = 3 sizeof ptri = 4 sizeof arrc[] = 3 sizeof ptrc = 4

B sizeof arri[] = 12 sizeof ptri = 4 sizeof arrc[] = 3 sizeof ptrc = 1

C sizeof arri[] = 3 sizeof ptri = 4 sizeof arrc[] = 3 sizeof ptrc = 1

D sizeof arri[] = 12 sizeof ptri = 4 sizeof arrc[] = 3 sizeof ptrc = 4



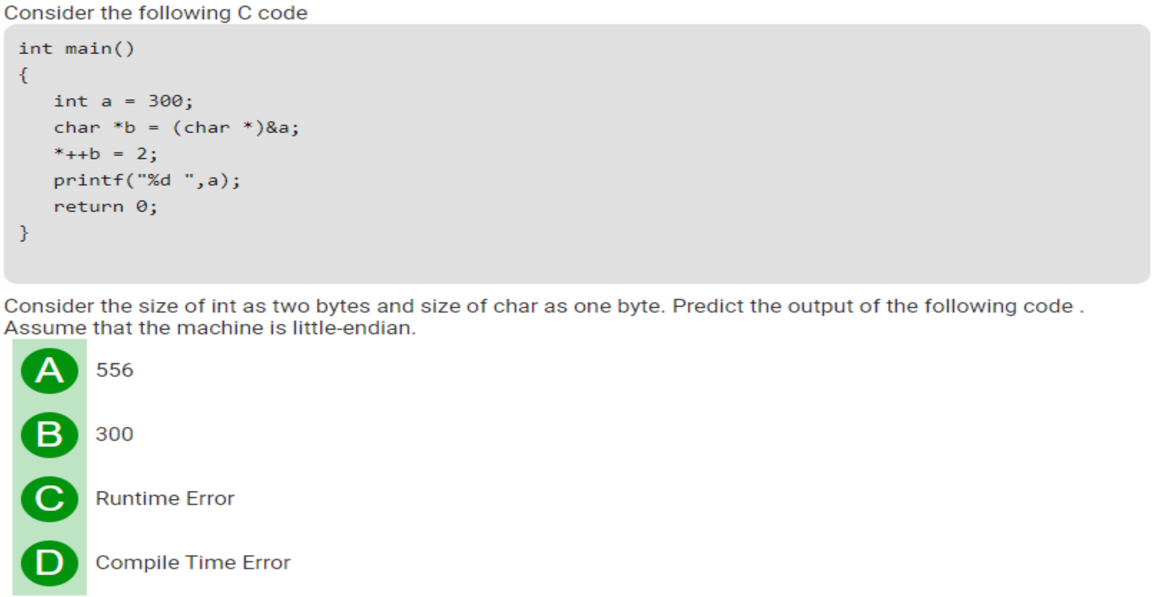
String Length using pointers

PROGRAM:

```
#include<stdio.h>
main()
{
   char s[25],*t;
   int len=0;
   printf("Enter a string\n");
   scanf("%s",&s);
   t=s;//copying base address of string

          Complet The Code Here

   printf("length of string is %d\n",len);
}
```

Output:

```
Enter a string
Hello
length of string is 5
```



Consider the following C code

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

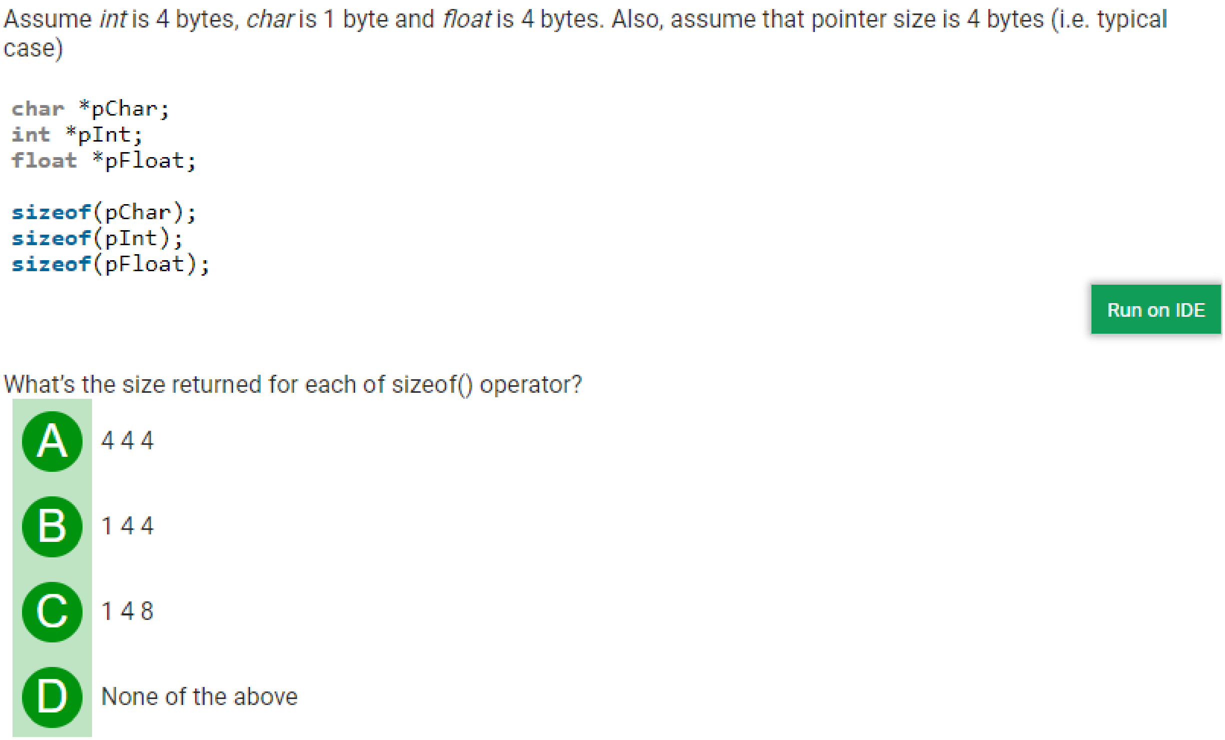
Consider the size of int as two bytes and size of char as one byte. Predict the output of the following code . Assume that the machine is little-endian.

A 556

B 300

C Runtime Error

D Compile Time Error



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);
```

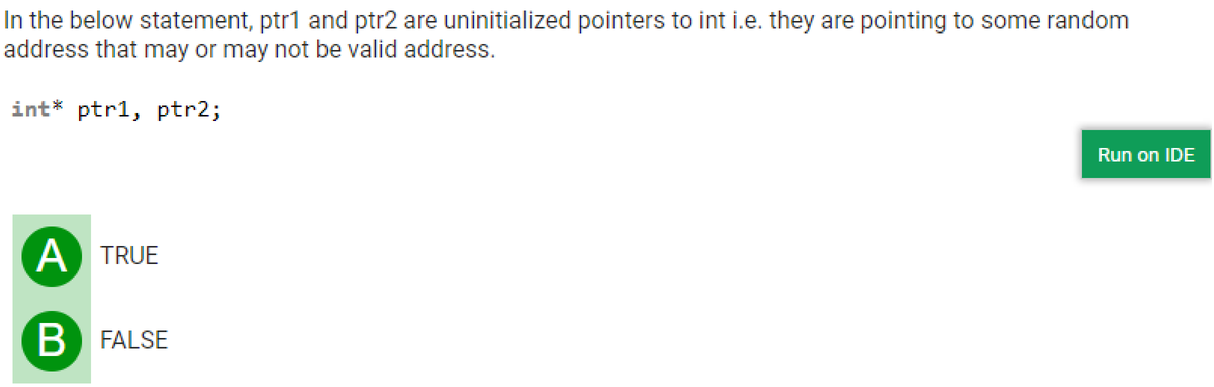
What's the size returned for each of sizeof() operator?

A 4 4 4

B 1 4 4

C 1 4 8

D None of the above



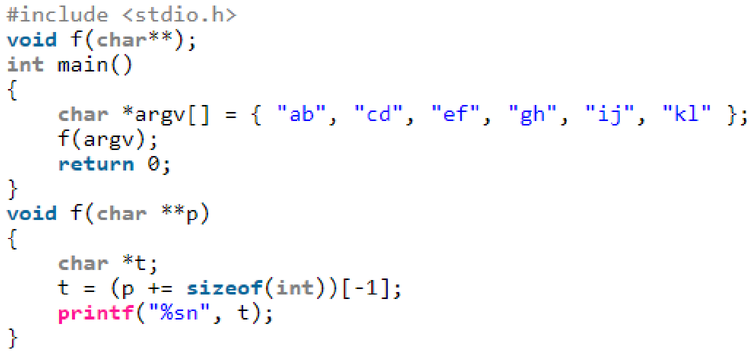
In the below statement, ptr1 and ptr2 are uninitialized pointers to int i.e. they are pointing to some random address that may or may not be valid address.

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

A TRUE

B FALSE

Assume that the size of int is 4.



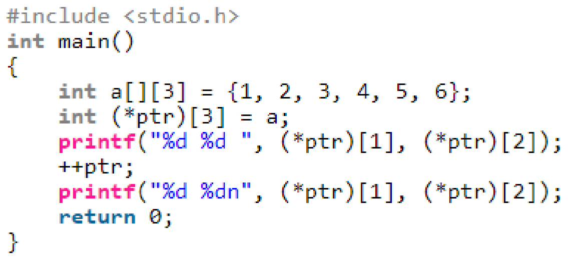
```c
#include <stdio.h>
void f(char**);
int main()
{
    char *argv[] = { "ab", "cd", "ef", "gh", "ij", "kl" };
    f(argv);
    return 0;
}
void f(char **p)
{
    char *t;
    t = (p += sizeof(int))[-1];
    printf("%sn", t);
}
```

A ab

B cd

C ef

D gh



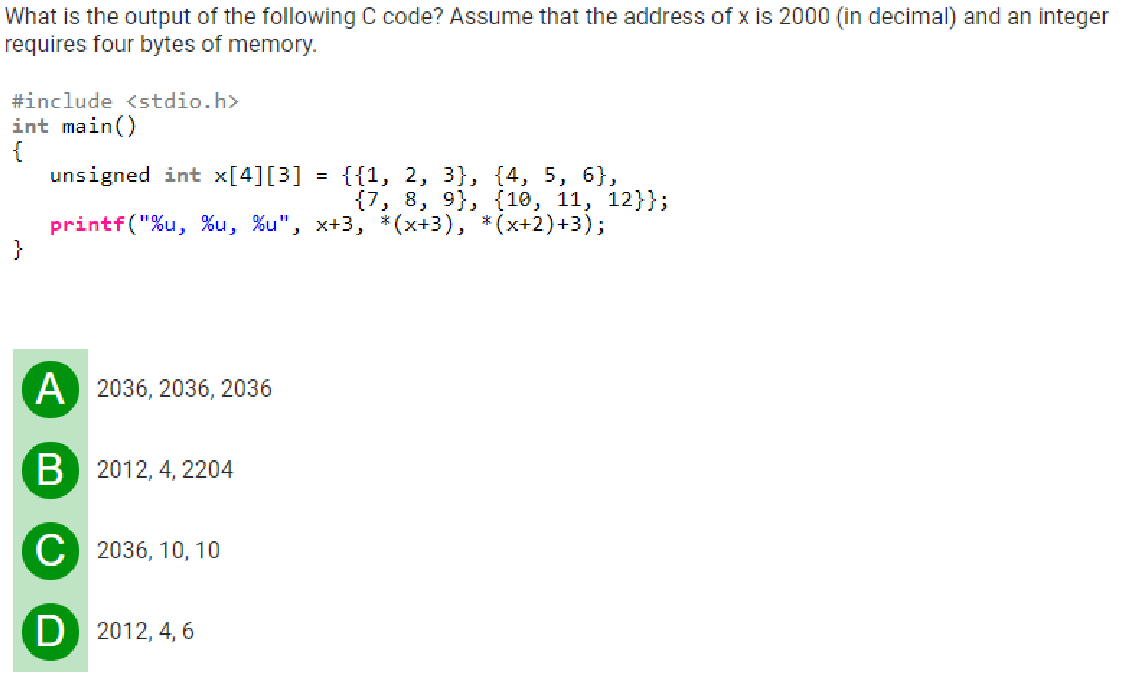
```c
#include <stdio.h>
int main()
{
    int a[][3] = {1, 2, 3, 4, 5, 6};
    int (*ptr)[3] = a;
    printf("%d %d ", (*ptr)[1], (*ptr)[2]);
    ++ptr;
    printf("%d %dn", (*ptr)[1], (*ptr)[2]);
    return 0;
}
```

A 2 3 5 6

B 2 3 4 5

C 4 5 0 0

D none of the above



What is the output of the following C code? Assume that the address of x is 2000 (in decimal) and an integer requires four bytes of memory.

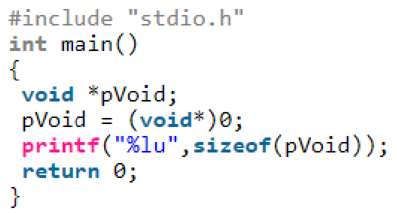
```
#include <stdio.h>
int main()
{
   unsigned int x[4][3] = {{1, 2, 3}, {4, 5, 6},
                           {7, 8, 9}, {10, 11, 12}};
   printf("%u, %u, %u", x+3, *(x+3), *(x+2)+3);
}
```

A 2036, 2036, 2036

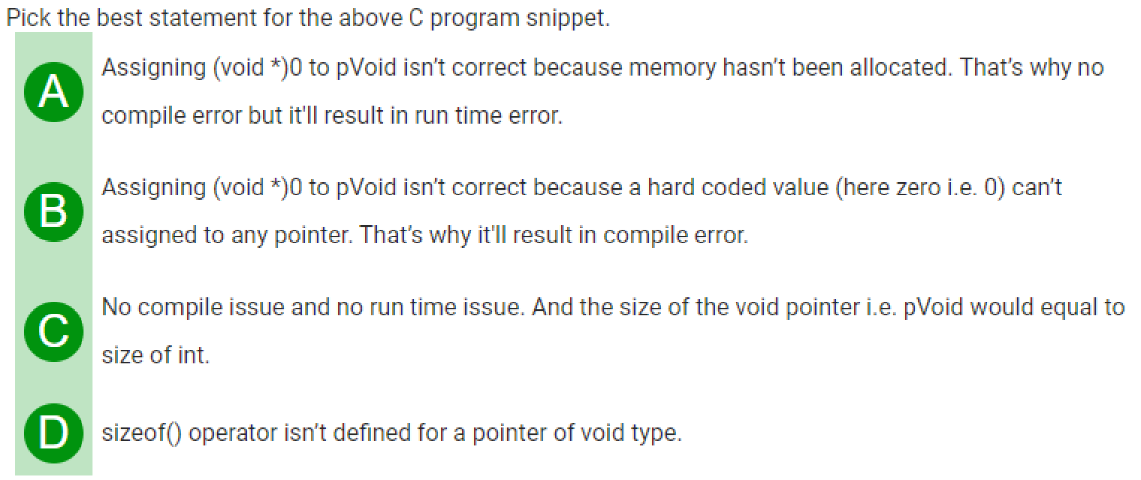
B 2012, 4, 2204

C 2036, 10, 10

D 2012, 4, 6



```
#include "stdio.h"
int main()
{
 void *pVoid;
 pVoid = (void*)0;
 printf("%lu",sizeof(pVoid));
 return 0;
}
```



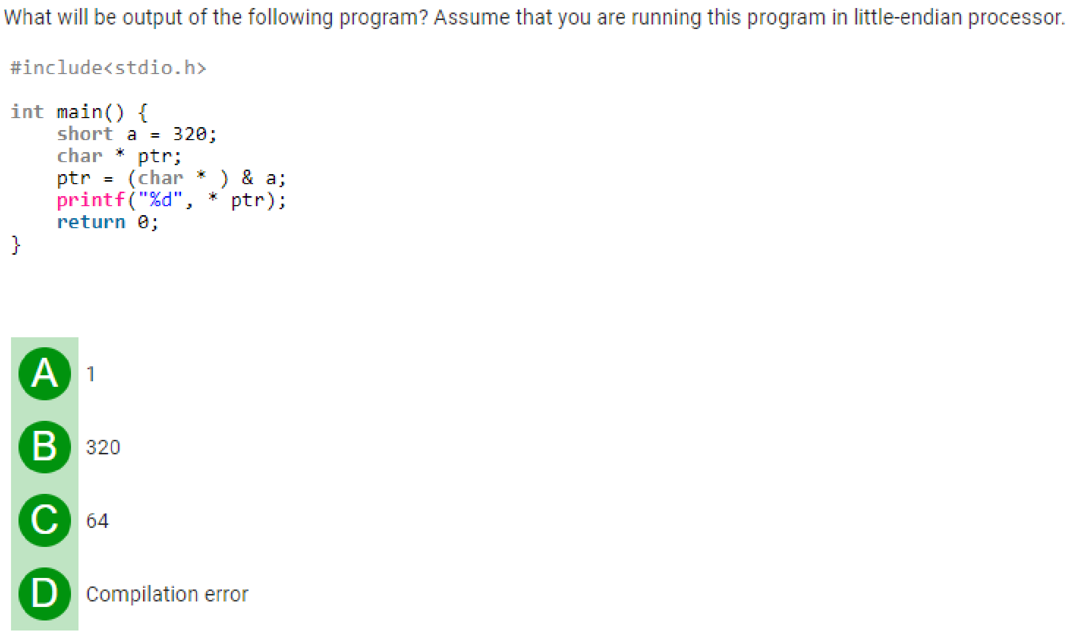
Pick the best statement for the above C program snippet.

A Assigning (void *)0 to pVoid isn't correct because memory hasn't been allocated. That's why no compile error but it'll result in run time error.

B Assigning (void *)0 to pVoid isn't correct because a hard coded value (here zero i.e. 0) can't assigned to any pointer. That's why it'll result in compile error.

C No compile issue and no run time issue. And the size of the void pointer i.e. pVoid would equal to size of int.

D sizeof() operator isn't defined for a pointer of void type.



What will be output of the following program? Assume that you are running this program in little-endian processor.

```
#include<stdio.h>

int main() {
    short a = 320;
    char * ptr;
    ptr = (char * ) & a;
    printf("%d", * ptr);
    return 0;
}
```

A 1

B 320

C 64

D Compilation error

2- What is ….

a- Int * ptr [10].

b- Double (*ptr)[38].

c- Short ** arr [5][10].

d- Struct book (*ptr[8]) (void).

3-It’s time for CODING:

1-

C Program to Count Number of Words in a given Text or Sentence
This is a C Program to Count the Number of Words in a given text or Sentence.

Problem Description
This program takes a string as input and count the number of words in the input string.

Note :

Recomended To Use this sentence to scan the string with the space without any problem

```
scanf("%[^\n]s", s);
```

2-

C Program to Accepts two Strings & Compare them
This is C program which accepts two strings & compare them.

Problem Description
This program accepts two strings as input and compares them.

1. **3-** Write a C program to copy one string to another using pointers.

4- Write a program in C to count the number of vowels and consonants in a string using a pointer.

5- Write a c code to make a calculator function that will take 3 argument and making summation for him but I will send the argument as any data type and the output will be as the input data type:
The key:
Void SUM (void * First , void * Second , int Flag);
If : Flag ==1
First and Second will return the int sum of him;
If : Flag ==2
First and Second will return the float sum of him;
If : Flag ==3
First and Second will return the double sum of him;

“MAKE IT WORK THEN OPTMIZE”