ABES Engineering College

Estd. 2000

College Code-032

Data Structure Training

Pointers

Array – One Dimensional

Department of CSE, CS & IT

ABES Engineering College, Ghaziabad

Data Structure Training

(Pointers)   P1

```
int a=10;
int *p;
p=&a;
```

A. `p contains an address of int.`

B. `value at address contained in p is an int`

C. `p is an int to pointer`

D. `p points to an int.`

Which of the following statements are correct? Choose the correct option.

i) A and B

ii) A and C

iii) A , B and D

iv) All

Data Structure Training

(Pointers) P1

```
int a=10;
int *p;
p=&a;
```

A. p contains an address of int.

B. value of address contained in p is an int

C. p is an int to pointer

D. p points to an int.

Which of the following statements are correct? Choose the correct option.

i) A and B

ii) A and C

iii) A , B and D

iv) All

Data Structure Training

(Pointers) P2
Use of * and &

TCS

```c
#include <stdio.h>
int x ;
int x = 11;
int main(void) {
int *ptr;
ptr = &x;
*ptr = 15;
printf(" x = %d\n", x);
printf(" *ptr = %d\n", *ptr);
return 0;
}
```

What would be the output of the above code ? Choose the correct option.

A. x = 15 , *ptr = 15

B. x = 11 , *ptr = 15

C. x = 0 , *ptr = 0

D. Compiler error

Data Structure Training

(Pointers) P2
Use of * and &

```c
#include <stdio.h>
int x ;
int x = 11;
int main(void) {
int *ptr;
ptr = &x;
*ptr = 15;
printf(" x = %d\n", x);
printf(" *ptr = %d\n", *ptr);
return 0;
}
```

What would be the output of the above code ? Choose the correct option.

[✓ **A.** x = 15 , *ptr = 15]

B. x = 11 , *ptr = 15

C. x = 0 , *ptr = 0

D. Compiler error

Data Structure Training

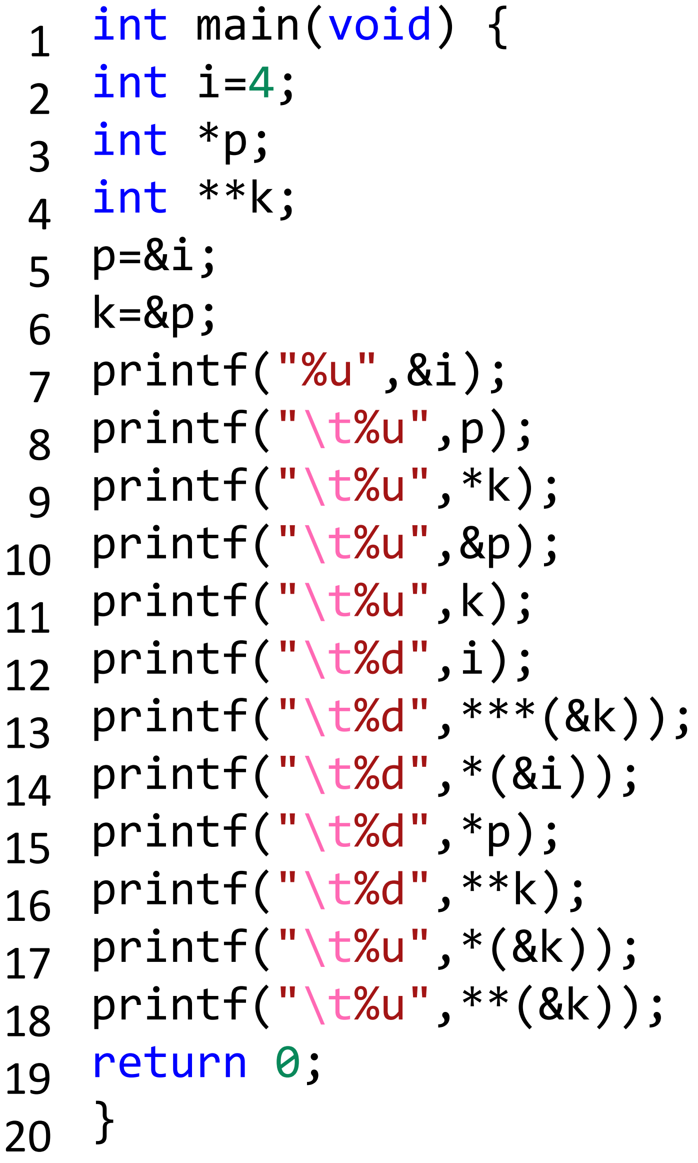
(Pointers) P3

Use of * and &

```c
int main(void)
{
int i=4;
int *p;
p=&i;
printf("%u",&i);
printf("\t%u",p);
printf("\t%d",i);
printf("\t%u",&p);
printf("\t%d",*p);
printf("\t%d",*(&i));
return 0;
}
```

Data Structure Training

(Pointers) P4
Use of * and &



```c
int main(void) {
int i=4;
int *p;
int **k;
p=&i;
k=&p;
printf("%u",&i);
printf("\t%u",p);
printf("\t%u",*k);
printf("\t%u",&p);
printf("\t%u",k);
printf("\t%d",i);
printf("\t%d",***(&k));
printf("\t%d",*(&i));
printf("\t%d",*p);
printf("\t%d",**k);
printf("\t%u",*(&k));
printf("\t%u",**(&k));
return 0;
}
```

Data Structure Training
(ARRAY)

What is Array?

- ✓ An array is **indexed collection** of **homogeneous** elements.
- ✓ There are three keywords in above definition-

1-**Indexed-**

Array elements are indexed and can be accessed by using their position. In most programming language index starts with zero.

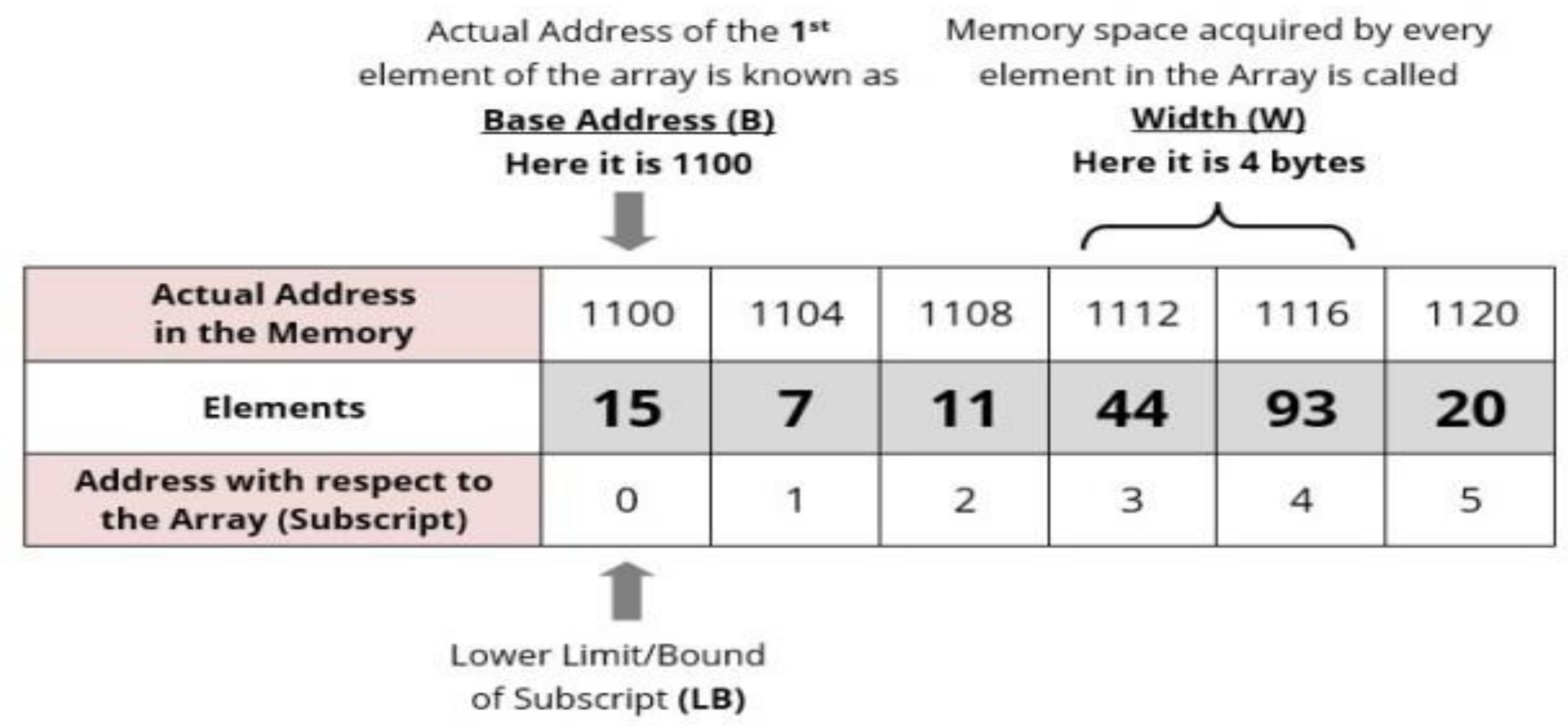
2-Collection-

An array is collection of elements.(more than one)

3-**Homogeneous-**

All elements in an array are of the same data type. It is only because of this homogeneity that we can access the array elements by using index.

Memory Representation of 1D Array



Actual Address of the 1st element of the array is known as **Base Address (B)**

Here it is 1100

Memory space acquired by every element in the Array is called **Width (W)**

Here it is 4 bytes

Actual Address in the Memory	1100	1104	1108	1112	1116	1120
Elements	**15**	**7**	**11**	**44**	**93**	**20**
Address with respect to the Array (Subscript)	0	1	2	3	4	5

Lower Limit/Bound of Subscript **(LB)**

Technical Training (**ARRAY**)

1- In C language data type, total number of elements and name of the array are provided while defining the array.

```
int arr[5]
```

2-

```
int arr[5]={1,2,3,4,5}; //initialization optional
```

3- Size of array can be skipped if initial value of all array elements are given.

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

4- If size of an array is more than the number of values in the initialization list, rest of the array gets initialized with zero.

```
int arr[1000]={0};
```

Pass arrays to a function in C

1. **Passing individual array elements**

```c
#include <stdio.h>
void display(int age1, int age2)
{
        printf("%d\n", age1);
        printf("%d\n", age2);
}
int main()
{
        int ageArray[] = {2, 8, 4, 12};
        display(ageArray[1], ageArray[2]);
        return 0;
}
```

Pass arrays to a function in C

2. Passing array element to function using Pointer

```c
#include <stdio.h>
void disp( int *num)
{
      printf("%d ", *num);
}
int main()
{
   int arr[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};
   for (int i=0; i<10; i++)
   {
     disp (&arr[i]);
   }
return 0;
}
```

Pass arrays to a function in C

3. Passing arrays to functions

```c
#include <stdio.h>
float calculateSum(float age[]);
int main()
{
        float result, age[] = {23.4, 55, 22.6, 3, 40.5, 18};
        result = calculateSum(age);
        printf("Result = %.2f", result);
        return 0;
}

float calculateSum(float age[])
{
        float sum = 0.0;
        for (int i = 0; i < 6; ++i)
        {
                sum += age[i];
        }
}
```

Pass arrays to a function in C

4. Passing entire array to function using Pointer

```c
#include <stdio.h>
void myfuncn( int *var1, int n)
{
    for(int x=0; x<n; x++)
    {
        printf("Value of var_arr[%d] is: %d \n", x, *var1);
        var1++;
    }
}

int main()
{
    int arr[] = {11, 22, 33, 44, 55, 66, 77};
    myfuncn(arr, 7);
    return 0;
}
```

Data Structure Training

(ARRAY) P5

```c
#include<stdio.h>
int main()
{
    int i;
    int arr[5] = {1};
    for (i = 0; i < 5; i++)
    printf("%d ", arr[i]);
    return 0;
}
```

What would be the output of given code? Choose the correct option.

A. 1 four garbage values.

B. 1 0 0 0 0

C. Compile error.

D. Runtime error.

Data Structure Training
(ARRAY) P5

```c
#include<stdio.h>
int main()
{
    int i;
    int arr[5] = {1};
    for (i = 0; i < 5; i++)
    printf("%d ", arr[i]);
    return 0;
}
```

What would be the output of given code? Choose the correct option.

A. 1 four garbage values.

[✓ **B.** 1 0 0 0 0]

C. Compile error.

D. Runtime error.

Data Structure Training

(ARRAY) P6

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

What would be the output of the above code ? Choose the correct option.

A. 1 1 1 1

B. 1 2 3 4

C. error

D. None

Data Structure Training

(ARRAY) P6

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

What would be the output of the above code ? Choose the correct option.

A. 1 1 1 1

B. 1 2 3 4

[✓ **C.** error]

D. None

Data Structure Training

(ARRAY) P7

```
#include <stdio.h>
int main(void)
{
int b[]={2,4,6,8,10};
int i;
for(i=0;i<5;i++)
{
*(b+i)=b[i]+i[b];
printf("%d ",*(i+b));
}
return 0;
}
```

What would be the output of the above code ? Choose the correct option.

A. 2 4 6 8 10

B. 4 8 12 16 20

C. error

D. 6 10 14 18 10

Data Structure Training

(ARRAY) P7

```c
#include <stdio.h>
int main(void)
{
int b[]={2,4,6,8,10};
int i;
for(i=0;i<5;i++)
{
*(b+i)=b[i]+i[b];
printf("%d ",*(i+b));
}
return 0;
}
```

What would be the output of the above code ? Choose the correct option.

A. 2 4 6 8 10

[✓ **B.** 4 8 12 16 20]

C. error

D. 6 10 14 18 10

Data Structure Training

(ARRAY) P8

```c
#include <stdio.h>
void change(int *p);
int main(void){
int b[5]={1,2,3,4,5};
int i;
change(b);
for(i=4;i>=0;i--)
{
printf("%d ",b[i]);
}
return 0;
}
void change(int *b){
int i;
for(i=0;i<5;i++){
*b=*b+1;
b++;
}
}
```

What would be the output of the above code ? Choose the correct option.

A. 5 4 3 2 1

B. 6 5 4 3 2

C. error

D. none

Data Structure Training

(ARRAY) P8

```c
#include <stdio.h>
void change(int *p);
int main(void){
int b[5]={1,2,3,4,5};
int i;
change(b);
for(i=4;i>=0;i--)
{
printf("%d ",b[i]);
}
return 0;
}
void change(int *b){
int i;
for(i=0;i<5;i++){
*b=*b+1;
b++;
}
}
```

What would be the output of the above code ? Choose the correct option.

A. 5 4 3 2 1

[✓ **B.** 6 5 4 3 2]

C. error

D. none

Data Structure Training

(ARRAY) P9

```c
#include <stdio.h>
int main(void) {
int n[20];
n[0]=100;
n[19]=120;
printf("%d %d",*n,*(n+19)+*(n+0));
return 0;
}
```

What would be the output of the above code ? Choose the correct option.

A. 100 220

B. 100 120

C. error

D. None

Data Structure Training

(ARRAY) P9

```c
#include <stdio.h>
int main(void) {
int n[20];
n[0]=100;
n[19]=120;
printf("%d %d",*n,*(n+19)+*(n+0));
return 0;
}
```

What would be the output of the above code ? Choose the correct option.

[✓ **A.** 100 220]

B. 100 120

C. error

D. None

Data Structure Training

(ARRAY) P10

```c
#include <stdio.h>
int main(void)
{
int b[]={1,2,3,4,5};
int j,*k;
k=&b[4]-4;
for(j=0;j<5;j++)
{
printf("%d ",*k);
k++;
}
return 0;
}
```

What would be the output of the above code ? Choose the correct option.

A. 1 1 1 1 1

B. 1 2 3 4 5

C. error

D. None

Data Structure Training

(ARRAY) P10

```
#include <stdio.h>
int main(void)
{
int b[]={1,2,3,4,5};
int j,*k;
k=&b[4]-4;
for(j=0;j<5;j++)
{
printf("%d ",*k);
k++;
}
return 0;
}
```

What would be the output of the above code ? Choose the correct option.

A. 1 1 1 1 1 [✓ B. 1 2 3 4 5]

C. error D. None

Data Structure Training

(ARRAY) P11

```c
#include <stdio.h>
int main(void)
{
int arr[]={1,2,3,4,5};
printf("%u %u",arr,&arr);
printf("\n%u %u",arr+1,&arr+1);
return 0;
}
```

What would be the output of the above code ? Choose the correct option. Assume array begins at address 200. Int data type is of size 4 bytes.

A. 200 200
204 220

B. 200 220
204 224

C. error

D. None

Data Structure Training

(ARRAY) P11

```c
#include <stdio.h>
int main(void)
{
int arr[]={1,2,3,4,5};
printf("%u %u",arr,&arr);
printf("\n%u %u",arr+1,&arr+1);
return 0;
}
```

What would be the output of the above code ? Choose the correct option. Assume array begins at address 200. Int data type is of size 4 bytes.

[✓ A. 200 200
204 220]

B. 200 220
204 224

C. error

D. None

Data Structure Training

(ARRAY) P12

```c
#include <stdio.h>
int main(){
    int a[3] = {1, 2, 3};
    int *p = a;
    int *r = &p;
    printf("%d", (**r));
    return(0);
}
```

What would be the output of the above code ? Choose the correct option.

A. 1

B. Compilation error

C. Runtime error

D. None

Data Structure Training

(ARRAY) P12

```c
#include <stdio.h>
int main(){
    int a[3] = {1, 2, 3};
    int *p = a;
    int *r = &p;
    printf("%d", (**r));
    return(0);
}
```

What would be the output of the above code ? Choose the correct option.

A. 1

[✓ B. Compilation error]

C. Runtime error

D. None

Data Structure Training

(ARRAY) P13

```c
# include <stdio.h>
void print(int arr[])
{
  int n = sizeof(arr)/sizeof(arr[0]);
  int i;
  for (i = 0; i < n; i++)
  printf("%d", arr[i]);
}
int main()
{
  int arr[] = {1, 2, 3, 4, 5, 6, 7, 8};
  print(arr);
  return 0;
}
```

Consider the following C program & Choose the correct option. Given size of int is 4 bytes and size of pointer data type is 8 bytes.

A. 1 2 3 4 5 6 7 8

B. 1 2 3 4

C. 1 2

D. 1

Data Structure Training

(ARRAY) P13

```c
# include <stdio.h>
void print(int arr[])
{
  int n = sizeof(arr)/sizeof(arr[0]);
  int i;
  for (i = 0; i < n; i++)
  printf("%d", arr[i]);
}
int main()
{
  int arr[] = {1, 2, 3, 4, 5, 6, 7, 8};
  print(arr);
  return 0;
}
```

Consider the following C program & Choose the correct option. Given size of int is 4 bytes and size of pointer data type is 8 bytes.

A. 1 2 3 4 5 6 7 8

B. 1 2 3 4

[✓ **C.** 1 2]

D. 1

Data Structure Training

(ARRAY) P14

```c
#include<stdio.h>
int main(){
    int i;
    int arr[5] = {1};
    for (i = 0; i <= 5; i++)
    printf("%d ", arr[i]);
    return 0;
}
```

What would be the output of the above code ? Choose the correct option.

A. Compiler Error: Array index out of bound.

B. The always prints 1 five times.

C. The program always crashes.

D. The program may print1 four times 0 followed by garbage value, or may crash if address (arr+5) is invalid.

Data Structure Training

(ARRAY) P14

```c
#include<stdio.h>
int main(){
    int i;
    int arr[5] = {1};
    for (i = 0; i <= 5; i++)
    printf("%d ", arr[i]);
    return 0;
}
```

What would be the output of the above code ? Choose the correct option.

A. Compiler Error: Array index out of bound.

B. The always prints 1 five times.

C. The program always crashes.

[✓ **D.** The program may print1 four times 0 followed by garbage value, or may crash if address (arr+5) is invalid.]

Data Structure Training

(ARRAY) P15

```c
#include <stdio.h>
void fun(int n)
{
	int idx;
	int arr1[n] = {0};
	int arr2[n];
	for (idx=0; idx<n; idx++)
	arr2[idx] = 0;
}
int main()
{
	fun(4);
	return 0;
}
```

What would be the output of the above code ? Choose the correct option.

A. Initialization of arr1 is incorrect. arr1 can’t be initialized due to its size being specified as variable. That’s why compile error.

B. Apart from definition of arr1 arr2, initialization of arr1 is also incorrect. arr1 can’t be initialized due to its size being specified as variable. That’s why compile error.

C. No error.

D. None of above.

Data Structure Training

(ARRAY) P15

```
#include <stdio.h>
void fun(int n)
{
        int idx;
        int arr1[n] = {0};
        int arr2[n];
        for (idx=0; idx<n; idx++)
        arr2[idx] = 0;
}
int main()
{
        fun(4);
        return 0;
}
```

What would be the output of the above code ? Choose the correct option.

[✓ **A.** Initialization of arr1 is incorrect. arr1 can’t be initialized due to its size being specified as variable. That’s why compile error.]

B. Apart from definition of arr1 arr2, initialization of arr1 is also incorrect. arr1 can’t be initialized due to its size being specified as variable. That’s why compile error.

C. No error.

D. None of above.

Data Structure Training

(ARRAY) P16

```
#include <stdio.h>
int main ()
{
        int i, j;
        int a [8] = {1, 2, 3, 4, 5, 6, 7, 8};
        for(i = 0; i < 3; i++)
        {
                a[i] = a[i] + 1;
                i++;
        }
        i--;
        for (j = 7; j > 4; j--)
        {
                i = j/2;
                a[i] = a[i] - 1;
        }
        printf ("%d, %d", i, a[i]);
        return(0);
}
```

What would be the output of the above code ? Choose the correct option.

A. 32

B. 23

C. 33

D. 22

Data Structure Training

(ARRAY) P16

```c
#include <stdio.h>
int main ()
{
        int i, j;
        int a [8] = {1, 2, 3, 4, 5, 6, 7, 8};
        for(i = 0; i < 3; i++)
        {
                a[i] = a[i] + 1;
                i++;
        }
        i--;
        for (j = 7; j > 4; j--)
        {
                i = j/2;
                a[i] = a[i] - 1;
        }
        printf ("%d, %d", i, a[i]);
        return(0);
}
```

What would be the output of the above code ? Choose the correct option.

A. 32

[✓ B. 23]

C. 33

D. 22

Data Structure Training

(ARRAY) P17

```c
#include<stdio.h>
void main()
{
        int ar[3]={1,2,3,4,5};
        for(int i=0;i<5;i++)
        {
                printf("%d ",ar[i]);
        }
}
```

What would be the output of the above code ? Choose the correct option.

A. 1 to 3 and 2 garbage values

B. 1 to 5

C. Compile error

D. Runtime error

Data Structure Training

(ARRAY) P17

```c
#include<stdio.h>
void main()
{
        int ar[3]={1,2,3,4,5};
        for(int i=0;i<5;i++)
        {
                printf("%d ",ar[i]);
        }
}
```

What would be the output of the above code ? Choose the correct option.

[✓ **A.** 1 to 3 and 2 garbage values]

B. 1 to 5

C. Compile error

D. Runtime error

Data Structure Training
(ARRAY)

Operations On Array:

1-Insertion

2-Deletion

3-Sorting

4-Searching

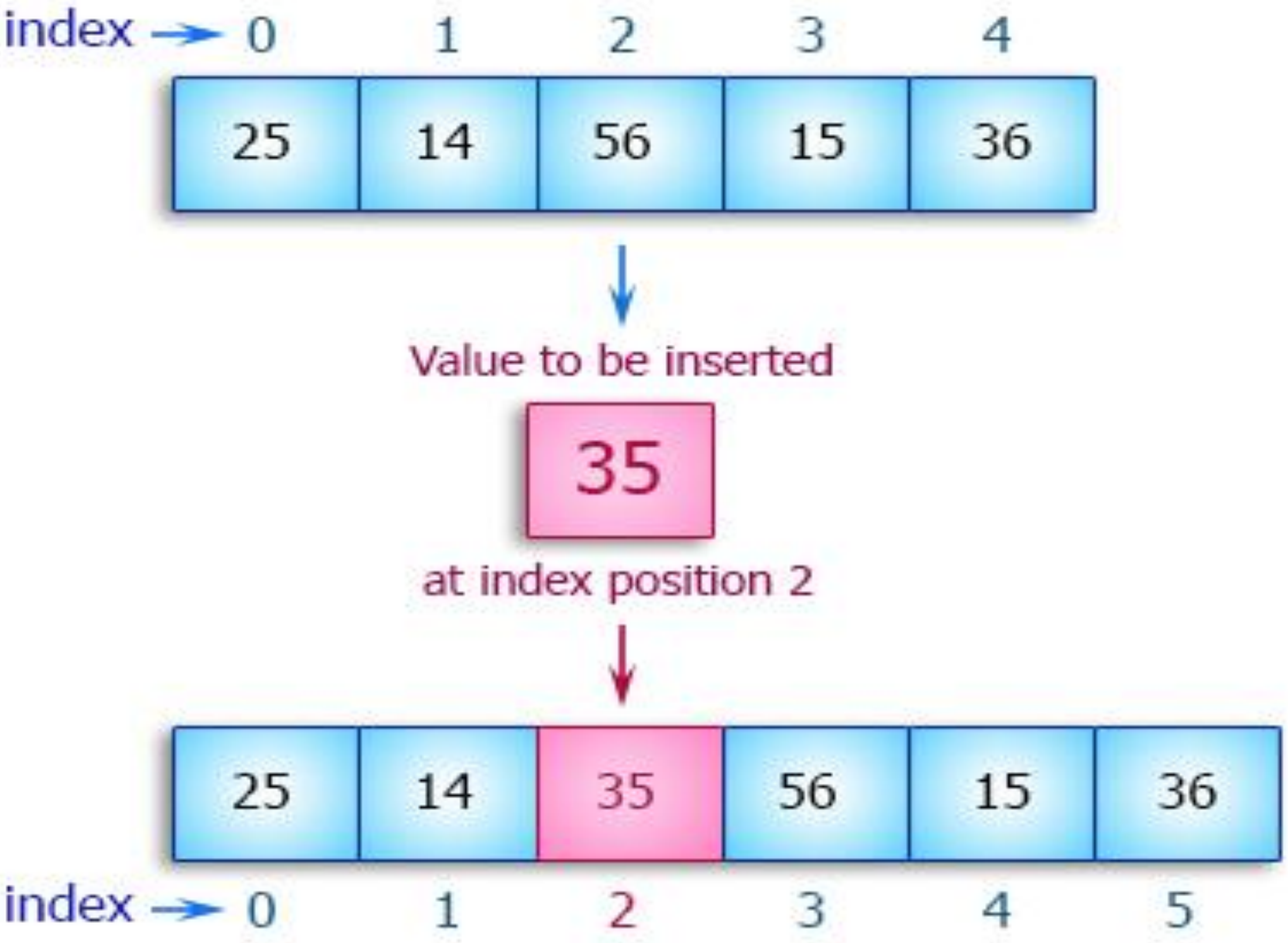
5-Merging

6-Traversing

Data Structure Training

(ARRAY)

Problem-1 Insert an element into an array by position.(Logic Development)



index →	0	1	2	3	4
	25	14	56	15	36

Value to be inserted

35

at index position 2

	25	14	35	56	15	36
index →	0	1	2	3	4	5

Data Structure Training

(ARRAY)

Problem-1 Insert an element into an array by position.(Logic Development)

```c
#include <stdio.h>
void insert_By_Position(int *arr,int pos,int item,int n);
int main(void)
{
        int arr[5]={1,2,3,4};
        insert_By_Position(arr,2,34,5);
        return 0;
}
void insert_By_Position(int *arr,int pos,int item,int n)
{
        int i;
        for (i=n-2;i>=pos;i--)
                // write your code here
        // write your code here
        for(i=0;i<n;i++)
        printf("%d\n",arr[i]);
}
```

Data Structure Training

(ARRAY)

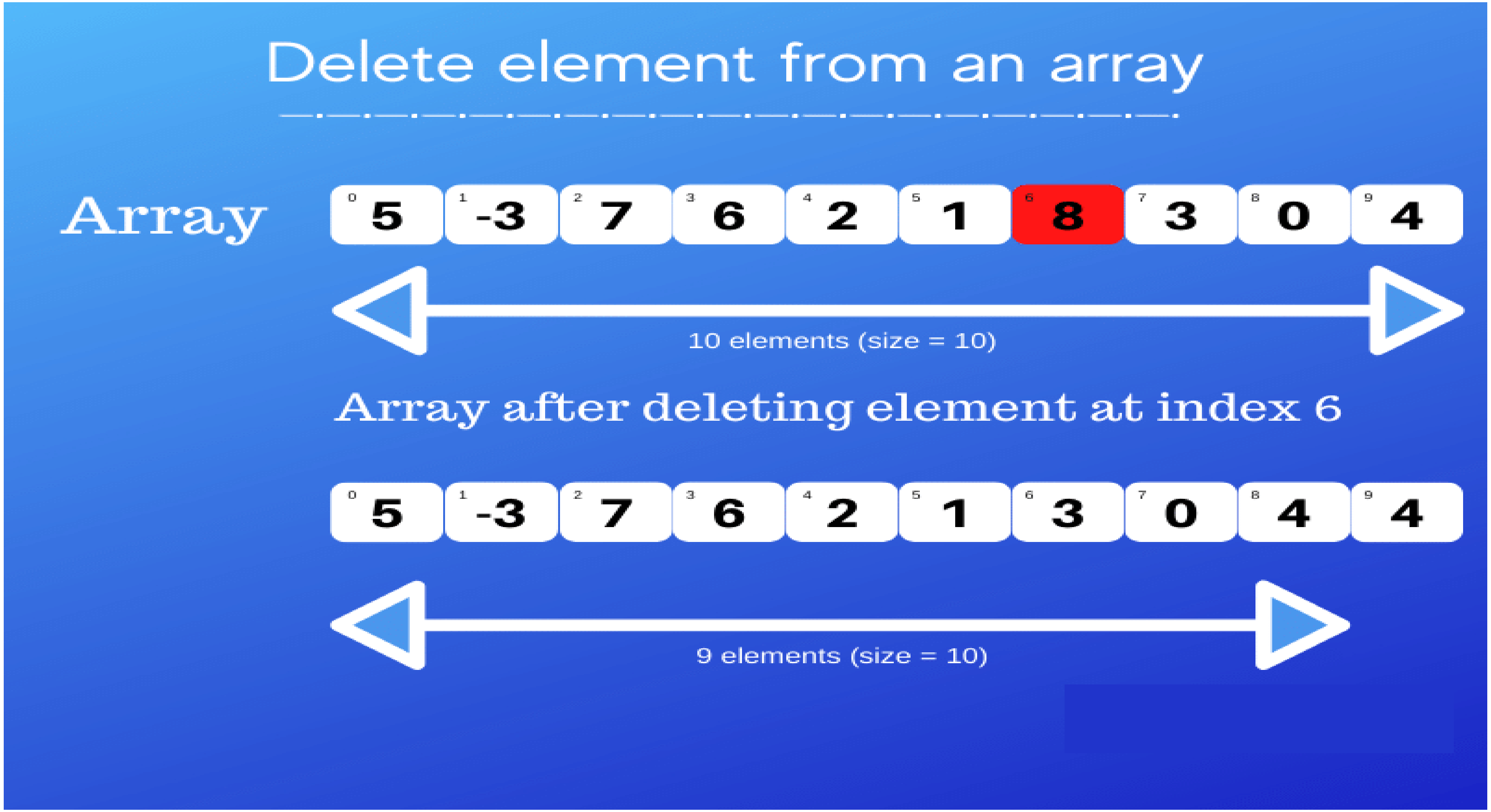
Problem-1 Insert an element into an array by position.(Implementation)

```
#include <stdio.h>
void insert_By_Position(int *arr,int pos,int item,int n);
int main(void)
{
        int arr[5]={1,2,3,4};
        insert_By_Position(arr,2,34,5);
        return 0;
}
void insert_By_Position(int *arr,int pos,int item,int n)
{
        int i;
        for (i=n-2;i>=pos;i--)
                a[i+1] = a[i];
        a[pos] = item;
        for(i=0;i<n;i++)
        printf("%d\n",arr[i]);
}
```

Data Structure Training

(ARRAY)

Problem-2 Delete an element at given Position from array.(Logic Development).



Delete element from an array

Array

0	1	2	3	4	5	6	7	8	9
5	-3	7	6	2	1	8	3	0	4

10 elements (size = 10)

Array after deleting element at index 6

0	1	2	3	4	5	6	7	8	9
5	-3	7	6	2	1	3	0	4	4

9 elements (size = 10)

Data Structure Training

(ARRAY)

Problem-2 Delete an element at given Position from array.(Implementation).

```
#include <stdio.h>
void deletion_By_Position(int *arr,int pos,int n);
int main(void)
 {
       int arr[5]={1,2,3,4,5};
       deletion_By_Position(arr,2,5);
       return 0;
}
void deletion_By_Position(int *arr,int pos,int n)
{
       int i;
       for (/*ADD STATEMENT*/)
             arr[i]=arr[i+1];
       /*ADD STATEMENT*/
       for(i=0;i<n;i++)
       printf("%d\n",arr[i]);
}
```

Data Structure Training

(ARRAY)

Problem-2 Delete an element at given Position from array.(Implementation).

```
#include <stdio.h>
void deletion_By_Position(int *arr,int pos,int n);
int main(void)
 {
      int arr[5]={1,2,3,4,5};
      deletion_By_Position(arr,2,5);
      return 0;
}
void deletion_By_Position(int *arr,int pos,int n)
{
      int i;
      for (i=pos-1; i<n-1; i++)
            arr[i]=arr[i+1];
      /*ADD STATEMENT*/
      for(i=0;i<n;i++)
      printf("%d\n",arr[i]);
}
```

Technical Training (**ARRAY**)

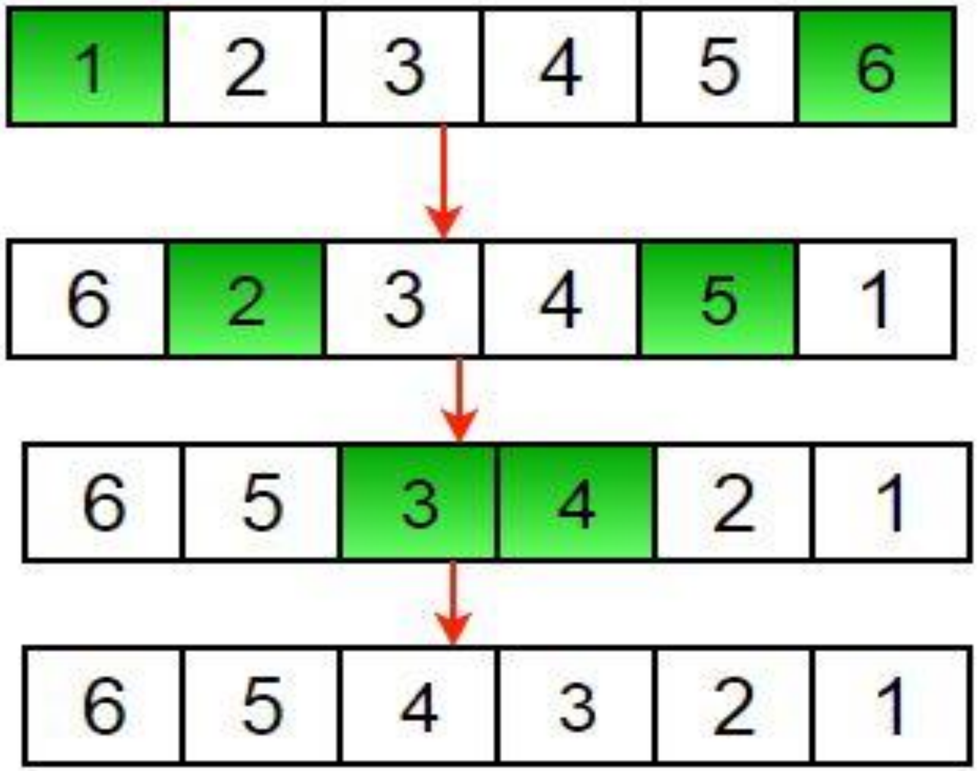
Problem-2 Delete an element at given Position from array.(Implementation).

```c
#include <stdio.h>
void deletion_By_Position(int *arr,int pos,int n);
int main(void)
 {
       int arr[5]={1,2,3,4,5};
       deletion_By_Position(arr,2,5);
       return 0;
}
void deletion_By_Position(int *arr,int pos,int n)
{
       int i;
       for (i=pos-1; i<n-1; i++)
             arr[i]=arr[i+1];
       n=n-1;
       for(i=0;i<n;i++)
       printf("%d\n",arr[i]);
}
```

Data Structure Training
(ARRAY)

Problem-3 How to reverse an array?

First Method- in this method we simply takes two variables low and high and set low=0 and high=n -1. Swap the elements of low and high and also after swapping increment low by one and decrement high by 1 till low less than high.

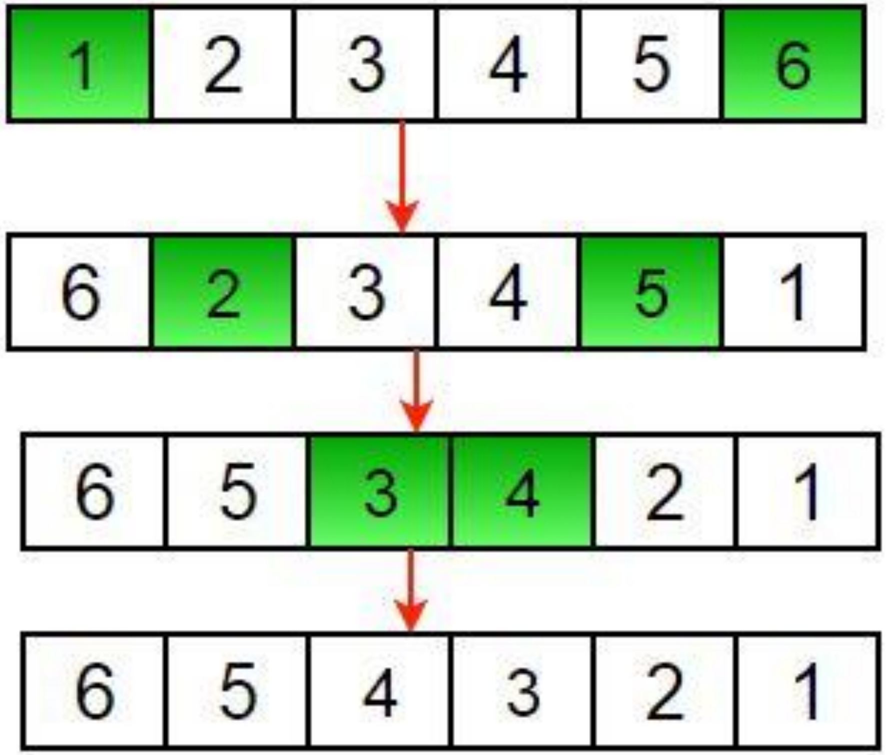


Data Structure Training
(ARRAY)

Problem-3 How to reverse an array?

First Method- in this method we simply takes two variables low and high and set low=0 and high=n -1.swap the elements of low and high and also after swapping increment low by one and decrement high by 1 till low less than high.

```
reverse(int *arr, int n)
{
        int low=0,high=n-1;
        while(low < high)
        {
                swap(&arr[low],&arr[high]);
                low++;
                high--;
        }
}
```



1 2 3 4 5 6

6 2 3 4 5 1

6 5 3 4 2 1

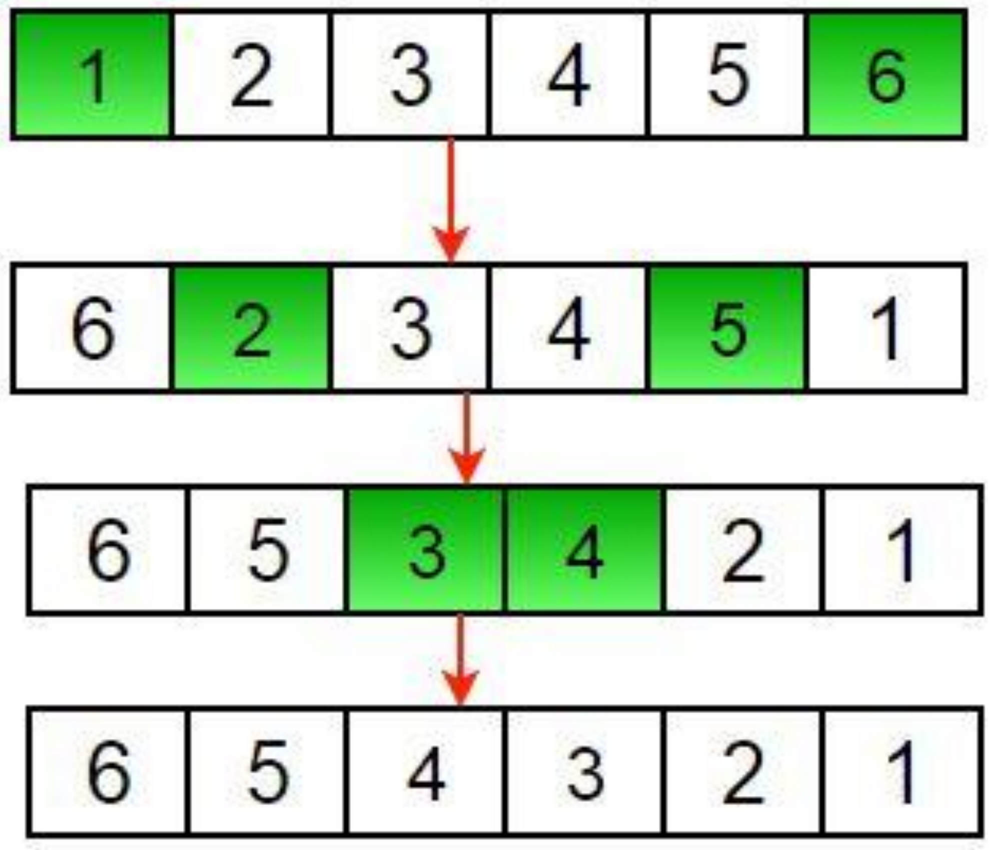
6 5 4 3 2 1

Data Structure Training

(ARRAY)

Problem-3 How to reverse an array?

Second Method- We can do the same thing by taking single variable.



1 2 3 4 5 6

6 2 3 4 5 1

6 5 3 4 2 1

6 5 4 3 2 1

Technical Training (ARRAY)

Problem-3 How to reverse an array?

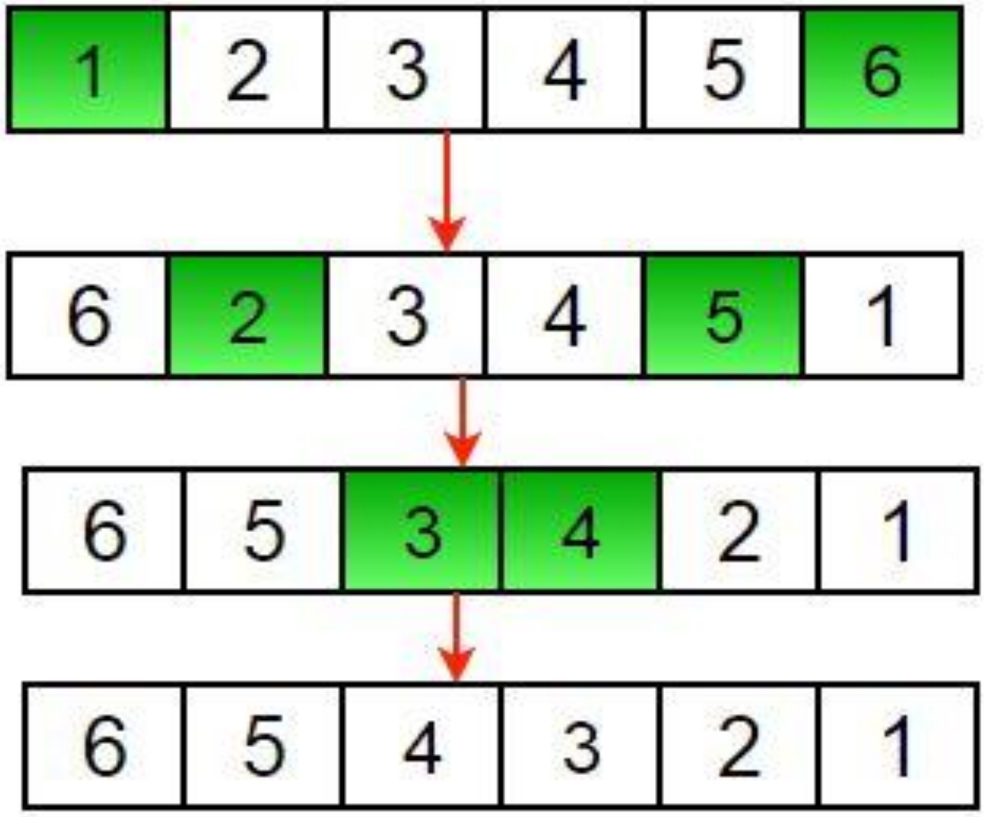
Second Method- We can do the same thing by taking single variable.

```
reverse(int *arr, int n)
{
       for(int i=0; i<n/2 ;i++)
             swap(&arr[i], &arr[n-1]);
}
void swap(int *a,int *b)
{
       int temp=*a;
       *a=*b;
       *b=temp;
}
```

Data Structure Training
(ARRAY)

Problem-3 How to reverse an array?

Third Method-(Recursion)- Recursive code takes more memory and time but recursion is such a powerful problem solving tool that every programmer should be very comfortable using it.

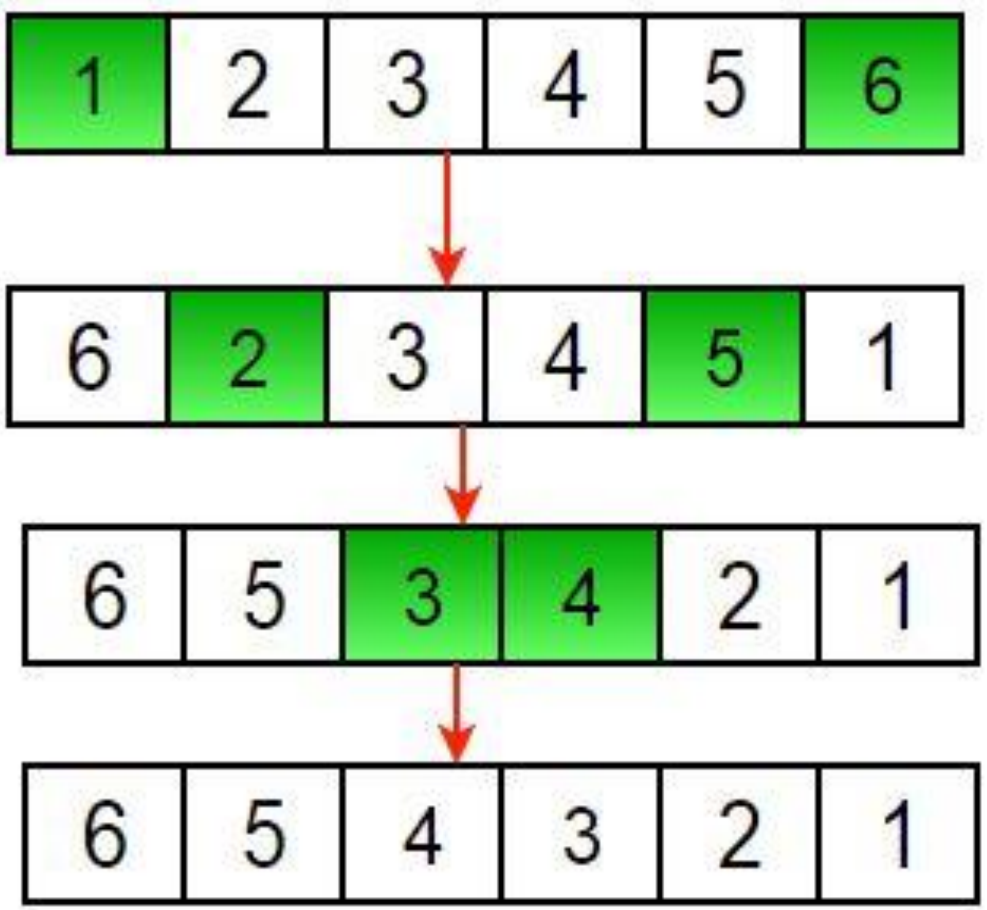


Data Structure Training
(ARRAY)

Problem-3 How to reverse an array?

Third Method-(Recursion) -Recursive code takes more memory and time but recursion is such a powerful problem solving tool that every programmer should be very comfortable using it.

```
reverse(int *arr, int low, int high)
{
        if (low>=high)
                return;
        swap(&arr[low],&arr[high]);
        reverse(arr,low+1,high-1);
}
```



1 2 3 4 5 6

6 2 3 4 5 1

6 5 3 4 2 1

6 5 4 3 2 1

Data Structure Training
(ARRAY)

```
1  #include<stdio.h>
2  void reverse(int *arr,int n);
3  void swap(int *a,int *b);
4  void reverse(int *arr, int n)
5  {
6    int i;
7    for( i=0; i<n/2 ;i++)
8    /*ADD STATEMENT*/
9    for(i=0;i<n;i++)
10   printf("%d\n",arr[i]);
11 }
12 void swap(int *a,int *b)
13 {
14
15   /* Write Code here */
16
17 }
18 int main()
19 {
20   int arr[6]={1,2,3,4,5,6};
21   reverse(arr,6);
22   return(0);
23 }
```

What should come at Line number 8?

Data Structure Training
(ARRAY)

```
1  #include<stdio.h>
2  void reverse(int *arr,int n);
3  void swap(int *a,int *b);
4  void reverse(int *arr, int n)
5  {
6    int i;
7    for( i=0; i<n/2 ;i++)
8    swap(&a[i], &a[n-i-1] );
9    for(i=0;i<n;i++)
10   printf("%d\n",arr[i]);
11 }
12 void swap(int *a,int *b)
13 {
14
15 /* Write Code here */
16
17  }
18 int main()
19 {
20   int arr[6]={1,2,3,4,5,6};
21   reverse(arr,6);
22   return(0);
23 }
```

What should come at Line numbers 14, 15 and 16?

Data Structure Training

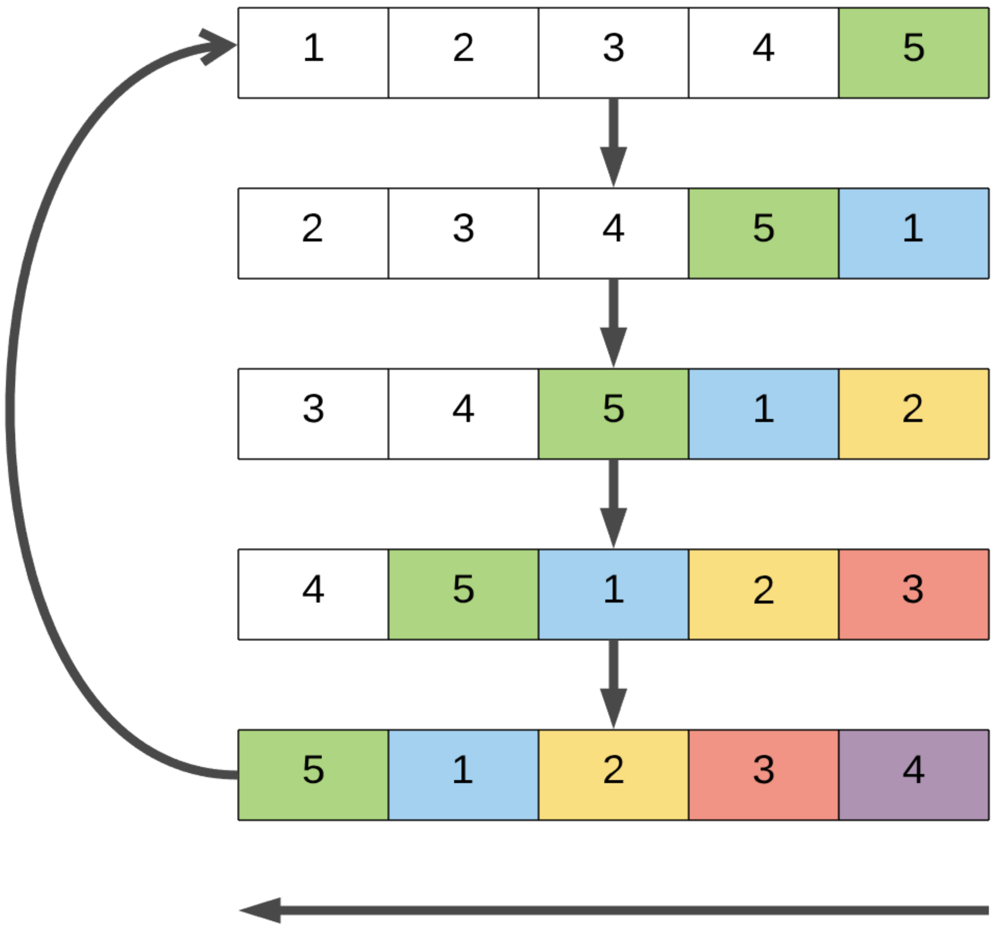
(ARRAY)

What should come at Line numbers 14, 15 and 16?

```
1   #include<stdio.h>
2   void reverse(int *arr,int n);
3   void swap(int *a,int *b);
4   void reverse(int *arr, int n)
5   {
6     int i;
7     for( i=0; i<n/2 ;i++)
8     swap(&a[i], &a[n-i-1] );
9     for(i=0;i<n;i++)
10    printf("%d\n",arr[i]);
11  }
12  void swap(int *a,int *b)
13  {
14     int temp = *a;
15    *a = *b;
16    *b = *temp;
17    }
18  int main()
19  {
20    int arr[6]={1,2,3,4,5,6};
21    reverse(arr,6);
22    return(0);
23  }
```

Data Structure Training
(ARRAY)

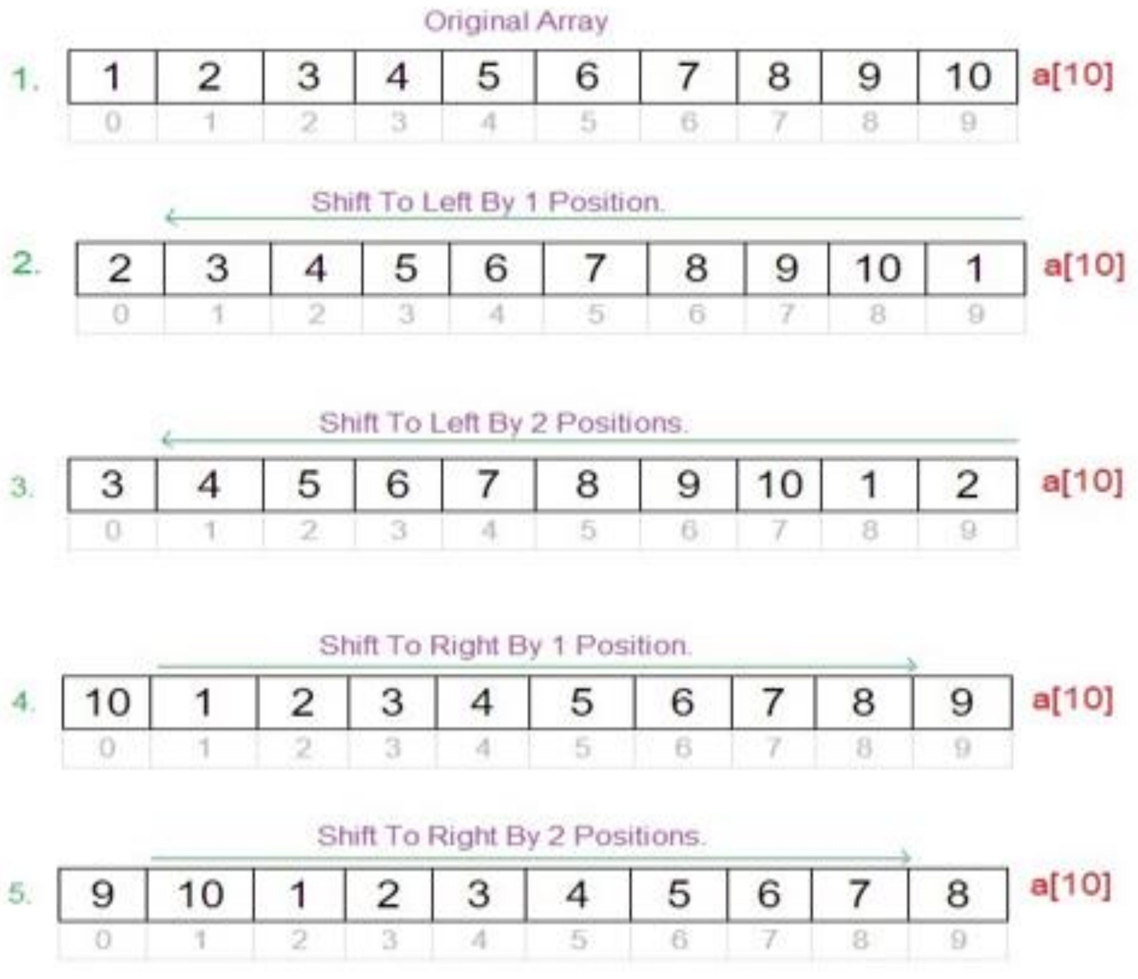
Problem-4 Rotate an array.



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

Flow of elements across rotations

Data Structure Training
(ARRAY)



Original Array

1. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | a[10]
0 1 2 3 4 5 6 7 8 9

Shift To Left By 1 Position.

2. | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | a[10]
0 1 2 3 4 5 6 7 8 9

Shift To Left By 2 Positions.

3. | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | a[10]
0 1 2 3 4 5 6 7 8 9

Shift To Right By 1 Position.

4. | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a[10]
0 1 2 3 4 5 6 7 8 9

Shift To Right By 2 Positions.

5. | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | a[10]
0 1 2 3 4 5 6 7 8 9

Problem-4 Rotate an array by position.
Given an array and a number d, how will you rotate an array by d positions.

Input : **int** a[]={1,2,3,4,5,6,7,8};
d=2

Output : 3,4,5,6,7,8,1,2

Data Structure Training
(ARRAY)

Problem-4 Rotate an array by position. Given an array and a number d, how will you rotate an array by d positions.

Input : **int** arr[]={1,2,3,4,5,6,7,8}; **d=2**

Output : 3,4,5,6,7,8,1,2

```
void rotate(int *arr,int d,int n)
{
        int i;
        for(i=1;i<=d;i++)
        {
        temp=arr[0];
                for(j=1;j<n;j++)
                {
                        arr[j-1]=arr[j];
                }
        arr[n-1]=temp;
        }
}
```

Data Structure Training

(ARRAY)

Problem-4 Rotate an array by position. Given an array and a number d, how will you rotate an array by d positions.

Input : **int** arr[]={1,2,3,4,5,6,7,8}; **d=2**

Output : 3,4,5,6,7,8,1,2

Method2-(use temporary array)

Data Structure Training
(ARRAY)

Problem-4 Rotate an array by position. Given an array and a number d, how will you rotate an array by d positions.

Input : **int** arr[]={1,2,3,4,5,6,7,8}; **d=2**

Output : 3,4,5,6,7,8,1,2

Method2-(use temporary array)

```
void rotate(int *arr,int n, int d)
{
        int i;
        int *temp=(int*)malloc(sizeof(int)*d);
        for(i=0;i<d;i++)
                temp[i]=arr[i];
        for(i=0;i<n-d;i++)
                arr[i]=arr[i+d];
        for( ;i<n;i++)
                arr[i]=temp[i-n+d];
        free(temp);

}
```

Data Structure Training

Array P18

```c
#include<stdio.h>
int arr[] = {5, 2, 3};
int main(void)
{
        int i;
        float arr[]={11.5, 2.6, 6.3, 9.1, 1.6};
        i = (arr + 1)[1];
        printf("%d", i);
        return 0;
}
```

What would be the output of the above code ? Choose the correct option.

A. 6

B. Error

C. 2

D. 9

Data Structure Training

Array P18

```
#include<stdio.h>
int arr[] = {5, 2, 3};
int main(void)
{
        int i;
        float arr[]={11.5, 2.6, 6.3, 9.1, 1.6};
        i = (arr + 1)[1];
        printf("%d", i);
        return 0;
}
```

What would be the output of the above code ? Choose the correct option.

[✓ **A.** 6] **B.** Error

C. 2 **D.** 9

Data Structure Training

Array P19

```
#include<stdio.h>
int arr[5];
int main(void)
{
	int i;
	arr[5] = {11, 2, 6, 9, 1, 7};
	i = (arr + 1)[1];
	printf("%d", i);
	return 0;
}
```

What would be the output of the above code ? Choose the correct option.

A.2 **B.** Error

C. 0 **D.** 11

Data Structure Training

Array P19

```c
#include<stdio.h>
int arr[5];
int main(void)
{
        int i;
        arr[5] = {11, 2, 6, 9, 1, 7};
        i = (arr + 1)[1];
        printf("%d", i);
        return 0;
}
```

What would be the output of the above code ? Choose the correct option.

A.2 [✓ B. Error]

C.0 D. 11

Data Structure Training

Array P20

```c
#include<stdio.h>
void fun(int arr[], size_t arr_size)
{
        int i;
        for (i = 0; i < arr_size; i++)
        {
                arr[i] = i/2;
        }
}

int main(void)
{
        int i;
        int arr[6] = {0, 11, 12, 3, 4, 'a'};
        fun(arr, 4);
        for(i = 0; i < sizeof(arr)/sizeof(double); i++)
        {
                printf("%d", arr[i]);
        }
return 0;
}
```

What would be the output of the above code ?
Choose the correct option.

A. 0 5 6 1 2 48

B. 0 5 6

C. 0 5 6 1

D. 0 0 1

Data Structure Training

Array P20

```
#include<stdio.h>
void fun(int arr[], size_t arr_size)
{
        int i;
        for (i = 0; i < arr_size; i++)
        {
                arr[i] = i/2;
        }
}

int main(void)
{
        int i;
        int arr[6] = {0, 11, 12, 3, 4, 'a'};
        fun(arr, 4);
        for(i = 0; i < sizeof(arr)/sizeof(double); i++)
        {
                printf("%d", arr[i]);
        }
return 0;
}
```

What would be the output of the above code ? Choose the correct option.

A. 0 5 6 1 2 48

B. 0 5 6

C. 0 5 6 1

[✓ D. 0 0 1]

Data Structure Training

(ARRAY) P21

Array Of Pointers-

```
#include <stdio.h>
int main(void)
{
      int a[]={10,20,30,40,50,60};
      int *b[]={a+3,a+4,a+5,a+2,a,a+1};
      int **c=b;
      c++;
      printf("%u",c-b);
      printf("%u",*c-a);
      printf("%%d",**c);
      return 0;
}
```

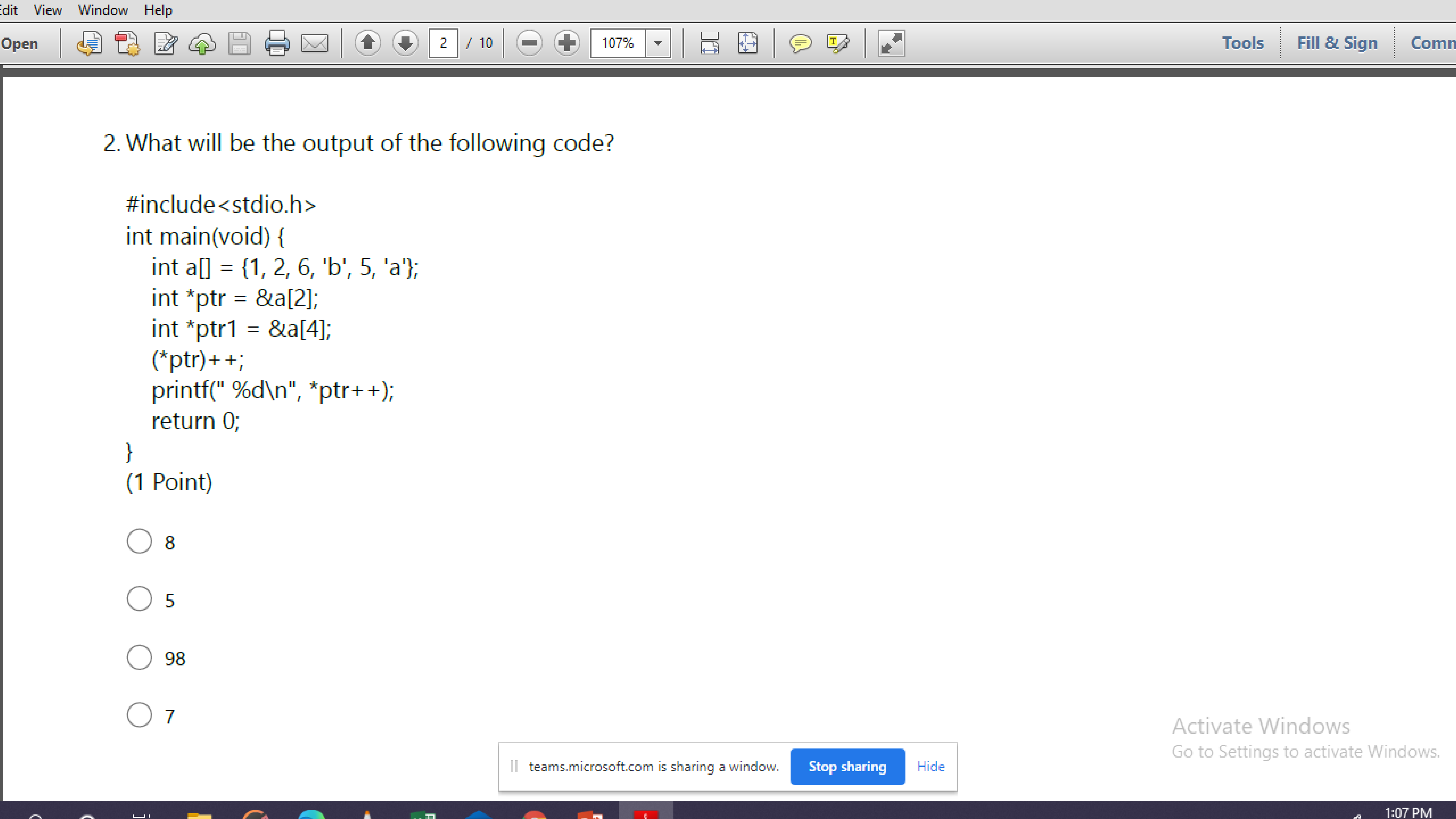
Data Structure Training

(ARRAY) P21

Array Of Pointers-

```
#include <stdio.h>
int main(void)
{
     int a[]={10,20,30,40,50,60};
     int *b[]={a+3,a+4,a+5,a+2,a,a+1};
     int **c=b;
     c++;
     printf("%u",c-b);    1
     printf("%u",*c-a);   4
     printf("%%d",**c);   50
     return 0;
}
```

Thank You



2. What will be the output of the following code?

```
#include<stdio.h>
int main(void) {
    int a[] = {1, 2, 6, 'b', 5, 'a'};
    int *ptr = &a[2];
    int *ptr1 = &a[4];
    (*ptr)++;
    printf(" %d\n", *ptr++);
    return 0;
}
```

(1 Point)

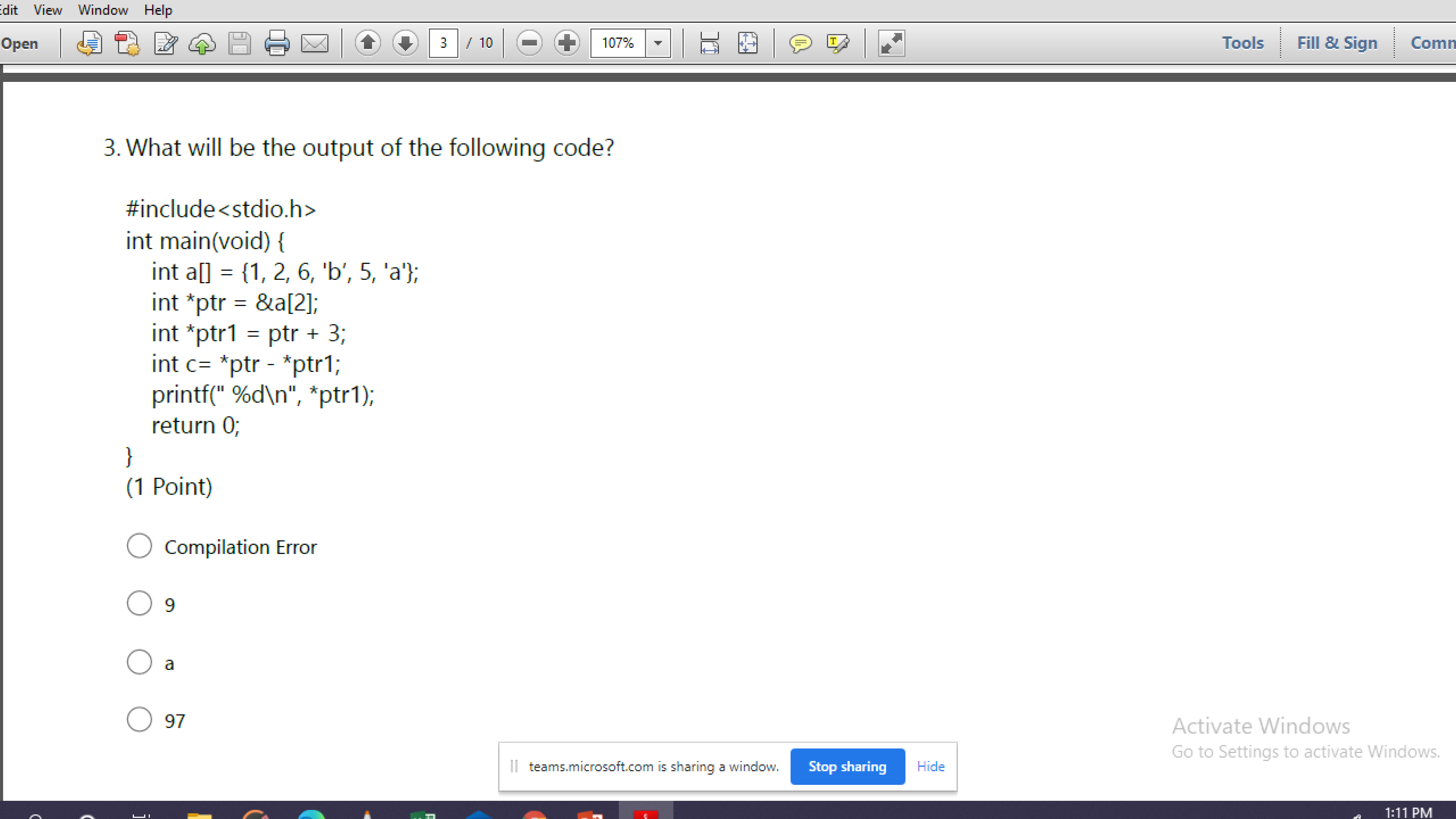
○ 8

○ 5

○ 98

○ 7

teams.microsoft.com is sharing a window.



3. What will be the output of the following code?

```
#include<stdio.h>
int main(void) {
    int a[] = {1, 2, 6, 'b', 5, 'a'};
    int *ptr = &a[2];
    int *ptr1 = ptr + 3;
    int c= *ptr - *ptr1;
    printf(" %d\n", *ptr1);
    return 0;
}
```

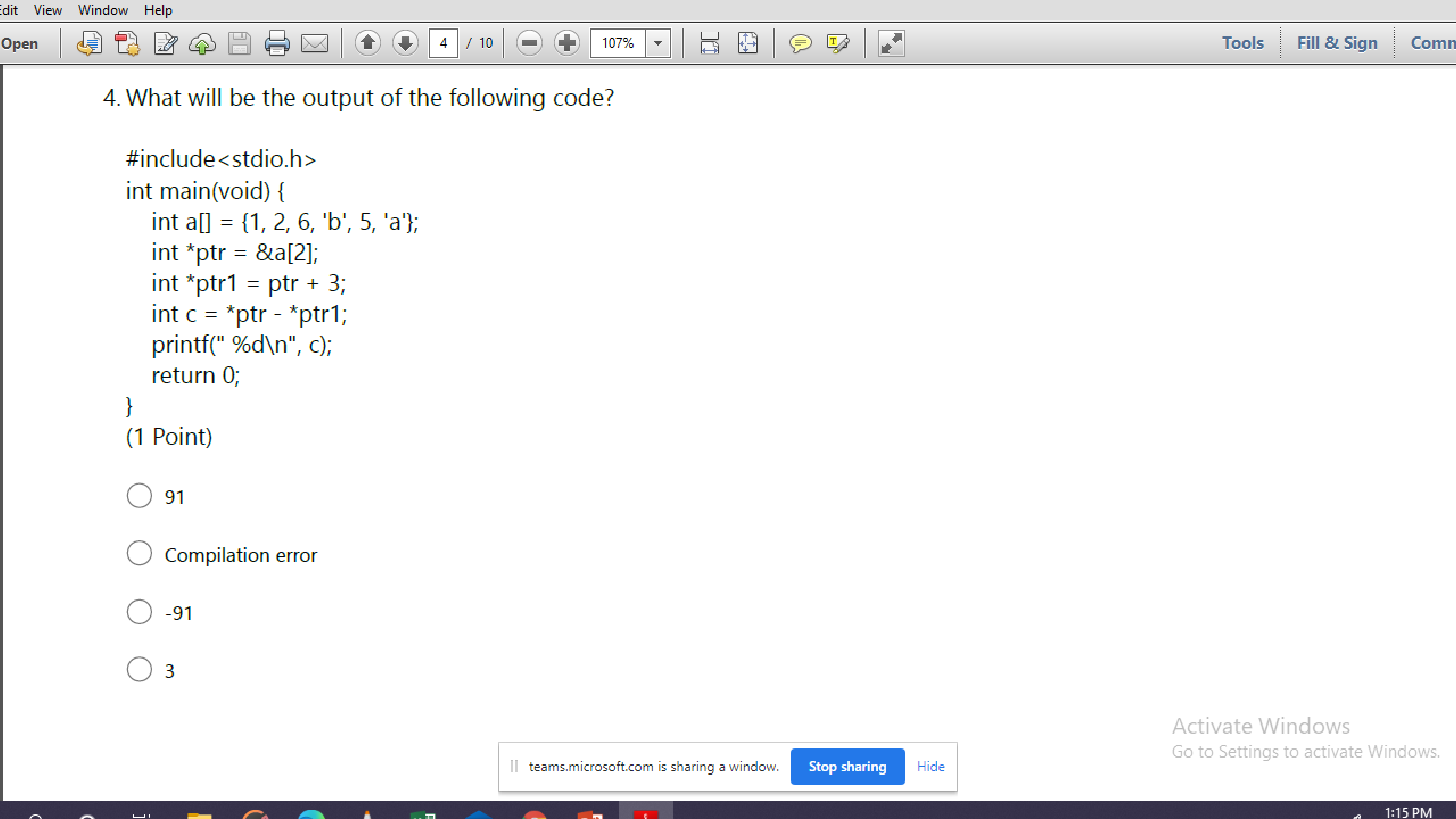
(1 Point)

○ Compilation Error

○ 9

○ a

○ 97



4. What will be the output of the following code?

```
#include<stdio.h>
int main(void) {
    int a[] = {1, 2, 6, 'b', 5, 'a'};
    int *ptr = &a[2];
    int *ptr1 = ptr + 3;
    int c = *ptr - *ptr1;
    printf(" %d\n", c);
    return 0;
}
```

(1 Point)

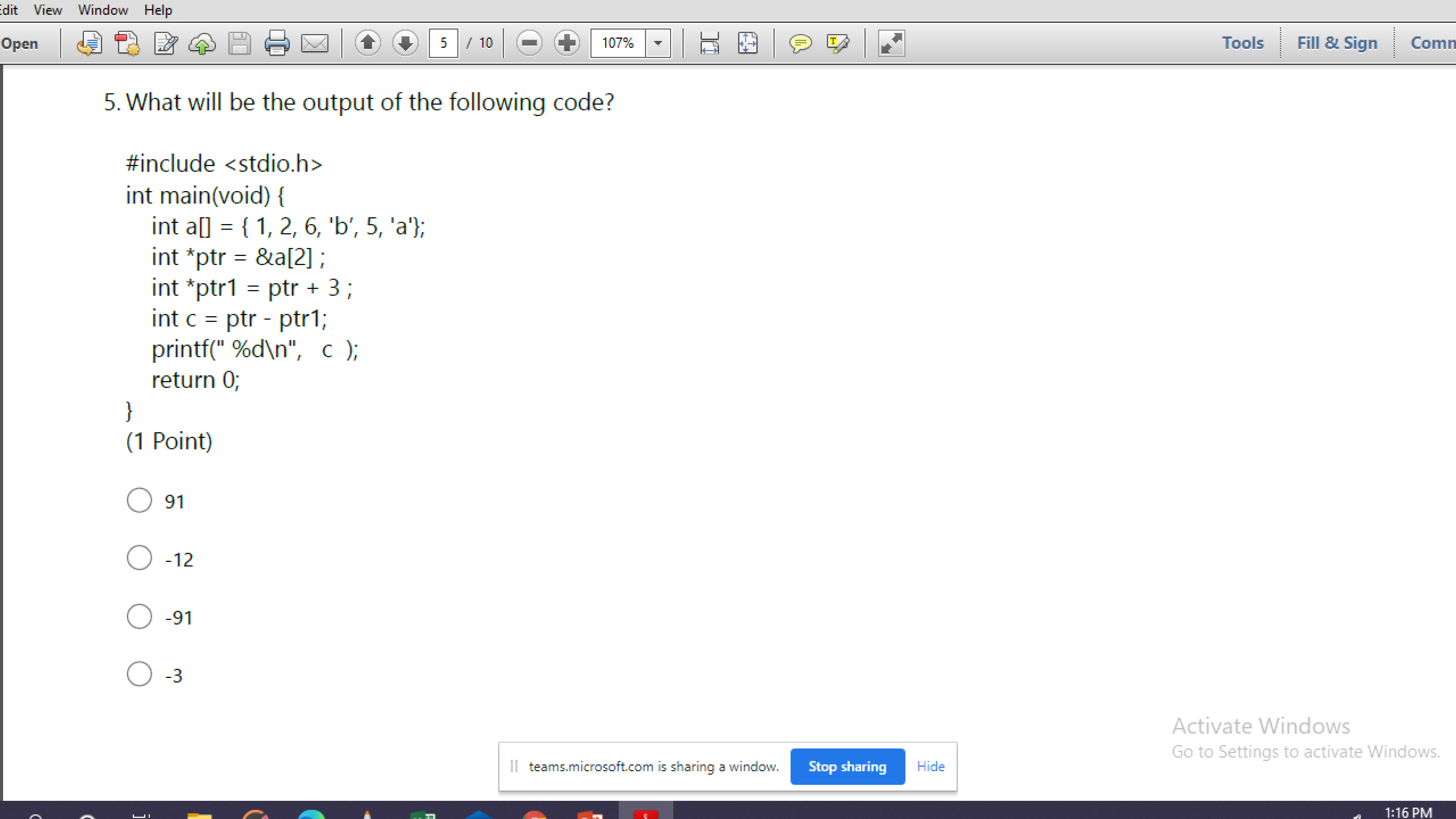
○ 91

○ Compilation error

○ -91

○ 3

teams.microsoft.com is sharing a window.



5. What will be the output of the following code?

```
#include <stdio.h>
int main(void) {
    int a[] = { 1, 2, 6, 'b', 5, 'a'};
    int *ptr = &a[2] ;
    int *ptr1 = ptr + 3 ;
    int c = ptr - ptr1;
    printf(" %d\n",   c  );
    return 0;
}
```

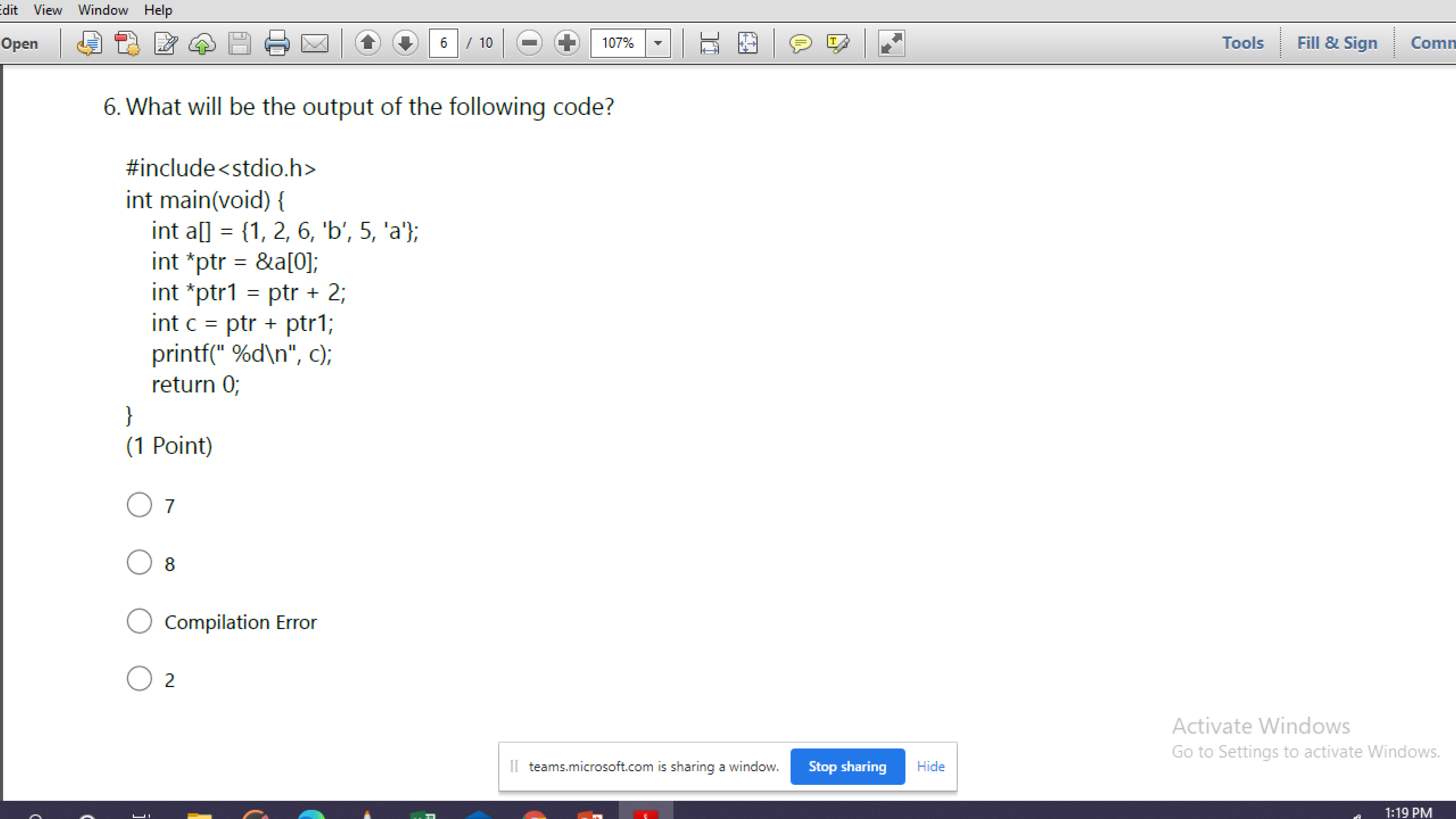
(1 Point)

○ 91

○ -12

○ -91

○ -3



6. What will be the output of the following code?

```
#include<stdio.h>
int main(void) {
    int a[] = {1, 2, 6, 'b', 5, 'a'};
    int *ptr = &a[0];
    int *ptr1 = ptr + 2;
    int c = ptr + ptr1;
    printf(" %d\n", c);
    return 0;
}
```

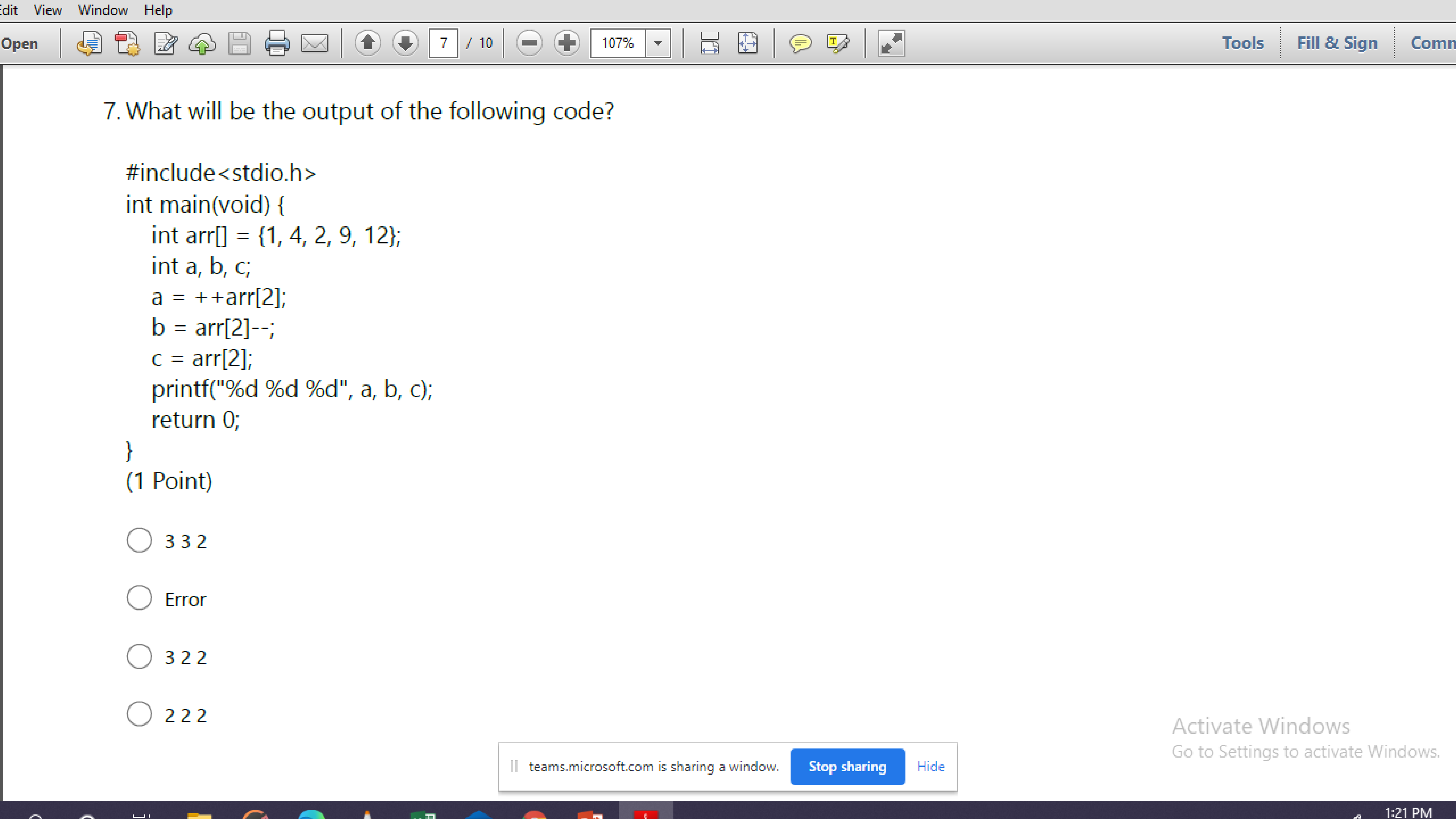
(1 Point)

○ 7

○ 8

○ Compilation Error

○ 2



7. What will be the output of the following code?

```
#include<stdio.h>
int main(void) {
    int arr[] = {1, 4, 2, 9, 12};
    int a, b, c;
    a = ++arr[2];
    b = arr[2]--;
    c = arr[2];
    printf("%d %d %d", a, b, c);
    return 0;
}
```

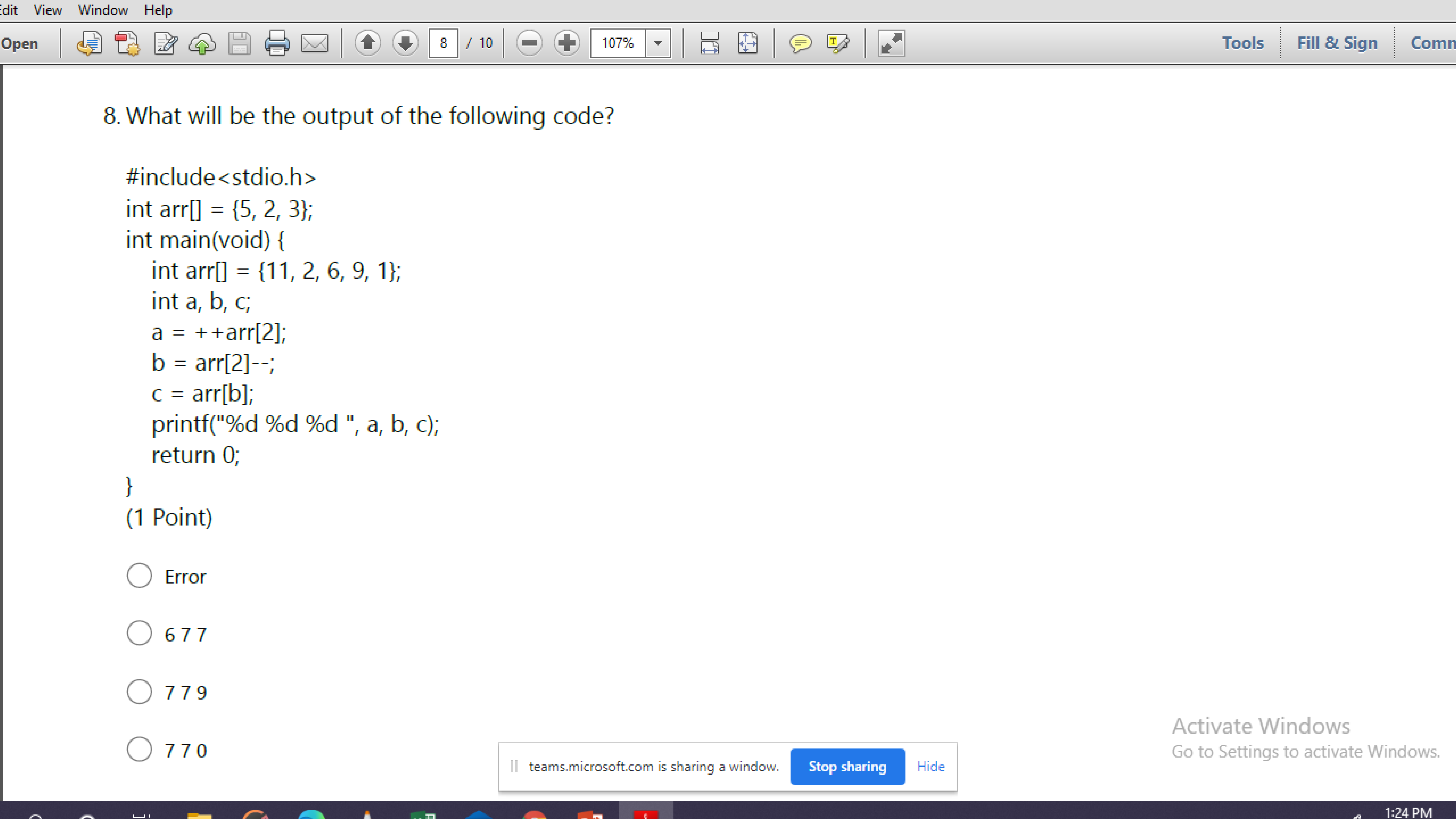
(1 Point)

○ 3 3 2

○ Error

○ 3 2 2

○ 2 2 2



8. What will be the output of the following code?

```
#include<stdio.h>
int arr[] = {5, 2, 3};
int main(void) {
    int arr[] = {11, 2, 6, 9, 1};
    int a, b, c;
    a = ++arr[2];
    b = arr[2]--;
    c = arr[b];
    printf("%d %d %d ", a, b, c);
    return 0;
}
```

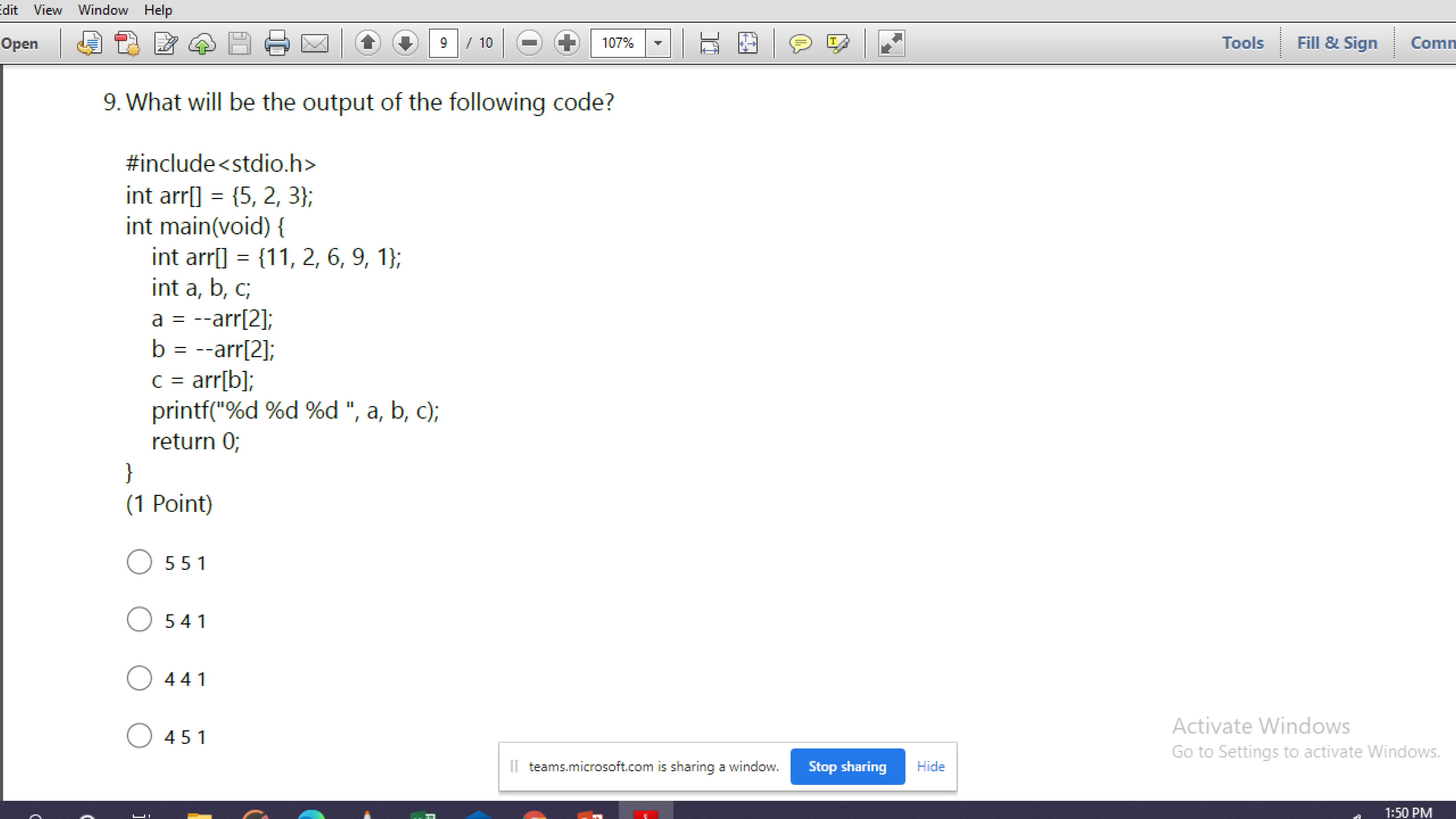
(1 Point)

○ Error

○ 6 7 7

○ 7 7 9

○ 7 7 0



9. What will be the output of the following code?

```
#include<stdio.h>
int arr[] = {5, 2, 3};
int main(void) {
    int arr[] = {11, 2, 6, 9, 1};
    int a, b, c;
    a = --arr[2];
    b = --arr[2];
    c = arr[b];
    printf("%d %d %d ", a, b, c);
    return 0;
}
```

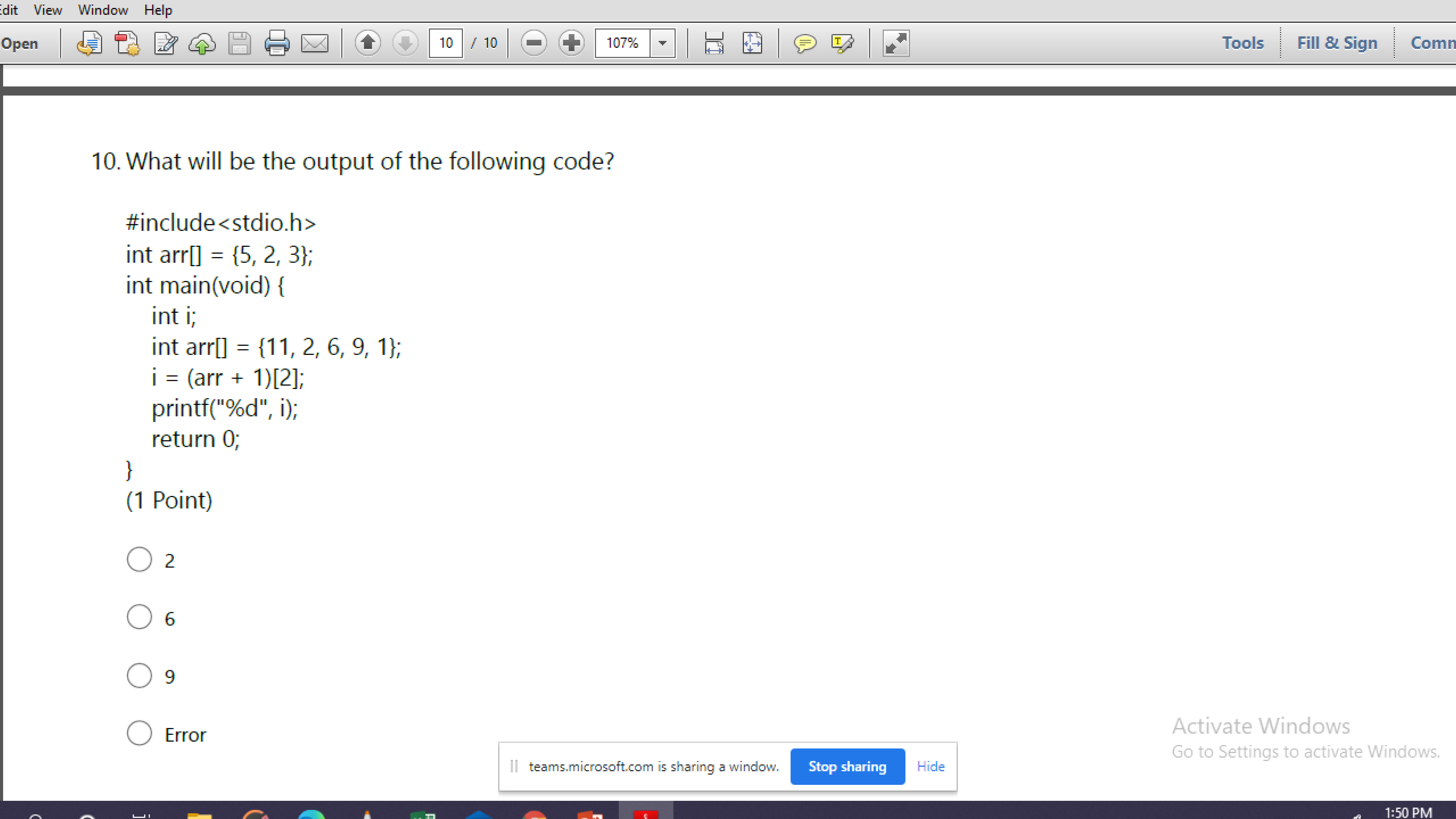
(1 Point)

○ 5 5 1

○ 5 4 1

○ 4 4 1

○ 4 5 1



10. What will be the output of the following code?

```
#include<stdio.h>
int arr[] = {5, 2, 3};
int main(void) {
    int i;
    int arr[] = {11, 2, 6, 9, 1};
    i = (arr + 1)[2];
    printf("%d", i);
    return 0;
}
```

(1 Point)

○ 2

○ 6

○ 9

○ Error