```
# include <stdio.h>
int main( )
{
int num[ 26 ], temp ;
num[ 0 ] = 100 ;
num[ 25 ] = 200 ;
temp = num[ 25 ] ;
num[ 25 ] = num[ 0 ] ;
num[ 0 ] = temp ;
printf ( "%d %d\n", num[ 0 ], num[ 25 ] ) ;
return 0 ;
}
```

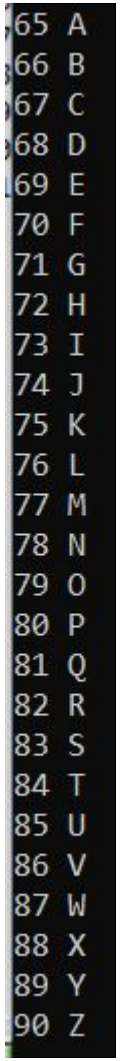
```
# include <stdio.h>
int main( )
{
int num[ 26 ], temp ;
num[ 0 ] = 100 ;
num[ 25 ] = 200 ;
temp = num[ 25 ] ;
num[ 25 ] = num[ 0 ] ;
num[ 0 ] = temp ;
printf ( "%d %d\n", num[ 0 ], num[ 25 ] ) ;
return 0 ;
}
```

Output:

```
200 100
```

```
# include <stdio.h>
int main( )
{
int array[ 26 ], i ;
for ( i = 0 ; i <= 25 ; i++ )
{
array[ i ] = 'A' + i ;
printf ( "%d %c\n", array[ i ], array[ i ] ) ;
}
return 0 ;
}
```

```c
# include <stdio.h>
int main( )
{
int array[ 26 ], i ;
for ( i = 0 ; i <= 25 ; i++ )
{
array[ i ] = 'A' + i ;
printf ( "%d %c\n", array[ i ], array[ i ] ) ;
}
return 0 ;
}
```



```
65 A
66 B
67 C
68 D
69 E
70 F
71 G
72 H
73 I
74 J
75 K
76 L
77 M
78 N
79 O
80 P
81 Q
82 R
83 S
84 T
85 U
86 V
87 W
88 X
89 Y
90 Z
```

```
# include <stdio.h>
int main( )
{
int sub[ 50 ], i ;
for ( i = 0 ; i <= 48 ; i++ ) ;
{
sub[ i ] = i ;
printf ( "%d\n", sub[ i ] ) ;
}
return 0 ;
}
```

```
# include <stdio.h>
int main( )
{
int sub[ 50 ], i ;
for ( i = 0 ; i <= 48 ; i++ ) ;
{
sub[ i ] = i ;
printf ( "%d\n", sub[ i ] ) ;
}
return 0 ;
}
```

Output:

```
49
```

Point out the errors, if any, in the following program segments:

```
# include <stdio.h>
int char mixed[ 100 ] ;
int main( )
{
int a[ 10 ], i ;
for ( i = 1 ; i <= 10 ; i++ )
{
scanf ( "%d", a[ i ] ) ;
printf ( "%d\n", a[ i ] ) ;
}
return 0 ;
}
```

Point out the errors, if any, in the following program segments:

```
# include <stdio.h>
int char mixed[ 100 ] ;
int main( )
{
int a[ 10 ], i ;
for ( i = 1 ; i <= 10 ; i++ )
{
scanf ( "%d", a[ i ] ) ;
printf ( "%d\n", a[ i ] ) ;
}
return 0 ;
}
```

Point out error

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

Point out error

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

Output

```
# include<stdio.h>
int main( )
{
int b[] = { 10, 20, 30, 40, 50 } ;
int i ;
for ( i = 0 ; i <= 4 ; i++ )
printf(" %d \n ", *( b+i ) ) ;
return 0 ;
}
```

Output

```
# include<stdio.h>
int main( )
{
int b[] = { 10, 20, 30, 40, 50 } ;
int i ;
for ( i = 0 ; i <= 4 ; i++ )
printf(" %d \n ", *( b+i ) ) ;
return 0 ;
}
```

```
Output:
10
20
30
40
50
```

```
# include <stdio.h>
int main( )
{
int b[ ] = { 0, 20, 0, 40, 5 } ;
int i, *k ;
k = b ;
for ( i = 0 ; i <= 4 ; i++ )
{
printf ( "%d\n" *k ) ;
k++ ;
}
return 0 ;
}
```

```
# include <stdio.h>
int main( )
{
int b[ ] = { 0, 20, 0, 40, 5 } ;
int i, *k ;
k = b ;
for ( i = 0 ; i <= 4 ; i++ )
{
printf ( "%d\n" *k ) ;
k++ ;
}
return 0 ;
}
```

```
Output:
0
20
0
40
5
```

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

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

Output:

```
7
9
11
 13
15
```

```
# include <stdio.h>
int main( )
{
static int a[ 5 ] ;
int i ;
for ( i = 0 ; i <= 4 ; i++ )
printf ( "%d\n", a[ i ] ) ;
return 0 ;
}
```

```
# include <stdio.h>
int main( )
{
static int a[ 5 ] ;
int i ;
for ( i = 0 ; i <= 4 ; i++ )
printf ( "%d\n", a[ i ] ) ;
return 0 ;
}
```

Output:

```
0
0
0
0
0
```

```
# include <stdio.h>
int main( )
{
int a[ 5 ] = { 5, 1, 15, 20, 25 } ;
int i, j, k = 1, m ;
i = ++a[ 1 ] ;
j = a[ 1 ]++ ;
m = a[ i++ ] ;
printf ( "%d %d %d\n", i, j, m ) ;
}
```

```
# include <stdio.h>
int main( )
{
int a[ 5 ] = { 5, 1, 15, 20, 25 } ;
int i, j, k = 1, m ;
i = ++a[ 1 ] ;
j = a[ 1 ]++ ;
m = a[ i++ ] ;
printf ( "%d %d %d\n", i, j, m ) ;
}
```

Output:

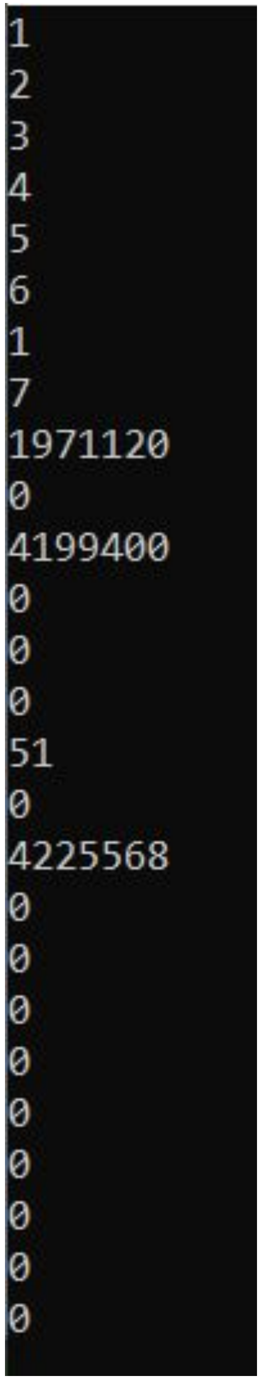
```
3 2 15
```

Error

```
# include <stdio.h>
int main( )
{
int array[ 6 ] = { 1, 2, 3, 4, 5, 6 } ;
int i ;
for ( i = 0 ; i <= 25 ; i++ )
printf ( "%d\n", array[ i ] ) ;
return 0 ;
}
```

Error

```
# include <stdio.h>
int main( )
{
int array[ 6 ] = { 1, 2, 3, 4, 5, 6 } ;
int i ;
for ( i = 0 ; i <= 25 ; i++ )
printf ( "%d\n", array[ i ] ) ;
return 0 ;
}
```



```
1
2
3
4
5
6
1
7
1971120
0
4199400
0
0
0
51
0
4225568
0
0
0
0
0
0
0
0
0
```

Error

```
int main( )
{
int a[ ] = { 10, 20, 30, 40, 50 } ;
int j ;
j = a ; /* store the address of zeroth element */
j = j + 3 ;
printf ( "%d\n",  *j ) ;
return 0 ;
}
```

error

```
int main( )
{
int a[ ] = { 10, 20, 30, 40, 50 } ;
int j ;
j = a ; /* store the address of zeroth element */
j = j + 3 ;
printf ( "%d\n",  *j ) ;
return 0 ;
}
```

J should have been declared a pointer variable

```
# include <stdio.h>
int main( )
{
float a[ ] = { 13.24, 1.5, 1.5, 5.4, 3.5 } ;
float *j ;
j = a ;
j = j + 4 ;
printf ( "%d %d %d\n", j, *j, a[ 4 ] ) ;
return 0 ;
}
```

```
# include <stdio.h>
int main( )
{
float a[ ] = { 13.24, 1.5, 1.5, 5.4, 3.5 } ;
float *j ;
j = a ;
j = j + 4 ;
printf ( "%d %d %d\n", j, *j, a[ 4 ] ) ;
return 0 ;
}
```

Output

```
6487568 0 0
```

```
# include <stdio.h>
int main( )
{
int max = 5 ;
float arr[ max ] ;
for ( i = 0 ; i < max ; i++ )
scanf ( "%f", &arr[ i ] ) ;
return 0 ;
}
```

```
# include <stdio.h>
int main( )
{
int max = 5 ;
float arr[ max ] ;
for ( i = 0 ; i < max ; i++ )
scanf ( "%f", &arr[ i ] ) ;
return 0 ;
}
```

Output
i not defined

```
# include <stdio.h>
int main( )
{
int n[ 3 ][ 3 ] = {
2, 4, 3,
6, 8, 5,
3, 5, 1
} ;
printf ( "%d %d %d\n", *n, n+1, *n+1 ) ;
return 0 ;
}
```

If address of n[0][0] is 6487536

```
# include <stdio.h>
int main( )
{
int n[ 3 ][ 3 ] = {
2, 4, 3,
6, 8, 5,
3, 5, 1
} ;
printf ( "%d %d %d\n", *n, n+1, *n+1 ) ;
return 0 ;
}
```

Output

```
6487536
6487548
6487540
```

```
# include <stdio.h>
int main( )
{
int n[ 3 ][ 3 ] = {
2, 4, 3,
6, 8, 5,
3, 5, 1
} ;
int i, *ptr ;
ptr = n ;
for ( i = 0 ; i <= 8 ; i++ )
printf ( "%d\n", *( ptr + i ) ) ;
return 0 ;
}
```

```
# include <stdio.h>
int main( )
{
int n[ 3 ][ 3 ] = {
2, 4, 3,
6, 8, 5,
3, 5, 1
} ;
int i, *ptr ;
ptr = n ;
for ( i = 0 ; i <= 8 ; i++ )
printf ( "%d\n", *( ptr + i ) ) ;
return 0 ;
}
```

Output

```
2
4
3
6
8
5
3
5
1
```

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

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

Output

```
2 2
4 4
3 3
6 6
8 8
5 5
3 3
5 5
1 1
```

Point out the errors, if any, in the following programs:

```
# include <stdio.h>
int main( )
{
int twod[ ][ ] = {
2, 4,
6, 8
} ;
printf ( "%d\n", twod ) ;
return 0 ;
}
```

Point out the errors, if any, in the following programs:

```
# include <stdio.h>
int main( )
{
int twod[ ][ ] = {
2, 4,
6, 8
} ;
printf ( "%d\n", twod ) ;
return 0 ;
}
```

Error

```
# include <stdio.h>
int main( )
{
int three[ 3 ][ ] = {
2, 4, 3,
6, 8, 2,
2, 3, 1
} ;
printf ( "%d\n", three[ 1 ][ 1 ] ) ;
return 0 ;
}
```

Error

```
# include <stdio.h>
int main( )
{
int three[ 3 ][ ] = {
2, 4, 3,
6, 8, 2,
2, 3, 1
} ;
printf ( "%d\n", three[ 1 ][ 1 ] ) ;
return 0 ;
}
```

```
# include <stdio.h>
int main( )
{
char c[ 2 ] = "A" ;
printf ( "%c\n", c[ 0 ] ) ;
printf ( "%s\n", c ) ;
return 0 ;
}
```

```
# include <stdio.h>
int main( )
{
char c[ 2 ] = "A" ;
printf ( "%c\n", c[ 0 ] ) ;
printf ( "%s\n", c ) ;
return 0 ;
}
```

```
A
A
```

```
# include <stdio.h>
int main( )
{
char s[ ] = "Get organised! learn C!!" ;
printf ( "%s\n", &s[ 2 ] ) ;
printf ( "%s\n", s ) ;
printf ( "%s\n", &s ) ;
printf ( "%c\n", s[ 2 ] ) ;
return 0 ;
}
```

```
# include <stdio.h>
int main( )
{
char s[ ] = "Get organised! learn C!!" ;
printf ( "%s\n", &s[ 2 ] ) ;
printf ( "%s\n", s ) ;
printf ( "%s\n", &s ) ;
printf ( "%c\n", s[ 2 ] ) ;
return 0 ;
}
```

Output:

```
t organised! learn C!!
Get organised! learn C!!
Get organised! learn C!!
t
```

```
# include <stdio.h>
int main( )
{
char s[ ] = "No two viruses work similarly" ;
int i = 0 ;
while ( s[ i ] != 0 )
{
printf ( "%c %c", s[ i ], *( s + i ) ) ;
printf ( "%c %c\n", i[ s ], *( i + s ) ) ;
i++ ;
}
return 0 ;
}
```

```
# include <stdio.h>
int main( )
{
char s[ ] = "No two viruses work similarly" ;
int i = 0 ;
while ( s[ i ] != 0 )
{
printf ( "%c %c", s[ i ], *( s + i ) ) ;
printf ( "%c %c\n", i[ s ], *( i + s ) ) ;
i++ ;
}
return 0 ;
}
```

Output

```
N N N N
o o o o
t t t t
w w w w
o o o o
v v v v
i i i i
r r r r
u u u u
s s s s
e e e e
s s s s
w w w w
o o o o
r r r r
k k k k
s s s s
i i i i
m m m m
i i i i
l l l l
a a a a
r r r r
l l l l
y y y y
```

```
# include <stdio.h>
int main( )
{
char s[ ] = "Churchgate: no church no gate" ;
char t[ 25 ] ;
char *ss, *tt ;
ss = s ;
while ( *ss != '\0' )
*tt++ = *ss++ ;
printf ( "%s\n", t ) ;
return 0 ;
}
```

```
# include <stdio.h>
int main( )
{
char s[ ] = "Churchgate: no church no gate" ;
char t[ 25 ] ;
char *ss, *tt ;
ss = s ;
while ( *ss != '\0' )
*tt++ = *ss++ ;
printf ( "%s\n", t ) ;
return 0 ;
}
```

No Output

```c
# include <stdio.h>
int main( )
{
char str1[] = { 'H', 'E', 'L', 'L', 'O' , '\0' } ;
char str2[] = "Hello" ;
printf ( "%s\n", str1 ) ;
printf ( "%s\n", str2 ) ;
return 0 ;
}
```

```c
# include <stdio.h>
int main( )
{
char str1[] = { 'H', 'E', 'L', 'L', 'O' , '\0' } ;
char str2[] = "Hello" ;
printf ( "%s\n", str1 ) ;
printf ( "%s\n", str2 ) ;
return 0 ;
}
```

OUTPUT

```
HELLO
Hello
```

```
# include <stdio.h>
int main( )
{
char str1[ ] = { 'H' , 'e', 'l', 'l', 'o', \0 } ;
char str2[ ] = "Hello" ;
printf ( "%s\n", str1 ) ;
printf ( "%s\n", str2 ) ;
return 0 ;
}
```

Error

```
# include <stdio.h>
void main( )
{
printf ( 5 + "Good Morning " ) ;
return 0 ;
}
```

```
# include <stdio.h>
void main( )
{
printf ( 5 + "Good Morning " ) ;
return 0 ;
}
```

Output

Morning

```
# include <stdio.h>
void main( )
{
printf ( "%c\n", "abcdefgh"[ 4 ] ) ;
return 0 ;
}
```

```
# include <stdio.h>
void main( )
{
printf ( "%c\n", "abcdefgh"[ 4 ] ) ;
return 0 ;
}
```

Output:

```
e
```

```
# include <stdio.h>
int main( )
{
printf ( "%d %d %d\n", sizeof ( '3' ), sizeof ( "3" ), sizeof ( 3 ) ) ;
return 0 ;
}
```

```
# include <stdio.h>
int main( )
{
printf ( "%d %d %d\n", sizeof ( '3' ), sizeof ( "3" ), sizeof ( 3 ) ) ;
return 0 ;
}
```

Output

```
4 2 4
```

‘3’ is a character constt . In c character constts have type int. thus size of ‘3’ = size of int

“3” is a string literal . It is a 2 element array –character 3 and NULL character

```
# include <stdio.h>
# include <string.h>
int main( )
{
char *str1 = "United" ;
char *str2 = "Front" ;
char *str3 ;
str3 = strcat ( str1, str2 ) ;
printf ( "%s\n", str3 ) ;
return 0 ;
}
```

```
# include <stdio.h>
# include <string.h>
int main( )
{
char *str1 = "United" ;
char *str2 = "Front" ;
char *str3 ;
str3 = strcat ( str1, str2 ) ;
printf ( "%s\n", str3 ) ;
return 0 ;
}
```

Output:

No output. Because no memory is allocated to str3, that is why it is pointing to NULL

IT WILL CAUSE A SEGMENTATION ERROR WHICH MEANS TRYING TO ACCESS MEMORY THAT DOES NOT BELONG TO YOU

```
# include <stdio.h>
int main( )
{
int arr[ ] = { 'A', 'B', 'C', 'D'  } ;
int i ;
for ( i = 0 ; i <= 3 ; i++ )
printf ( "%d\t", arr[ i ] ) ;
printf ( "\n" ) ;
return 0 ;
}
```

```c
# include <stdio.h>
int main( )
{
int arr[ ] = { 'A', 'B', 'C', 'D'  } ;
int i ;
for ( i = 0 ; i <= 3 ; i++ )
printf ( "%d\t", arr[ i ] ) ;
printf ( "\n" ) ;
return 0 ;
}
```

Output:

```
65      66      67      68
```

EXERCISE

Match the following with reference to the following program segment:

```
int x[ 3 ][ 5 ] = {
{ 1, 2, 3, 4, 5 },
{ 6, 7, 8, 9, 10 },
{ 11, 12, 13, 14, 15 }
}, *n = &x ;
```

1. *(*(x + 2) + 1)
2. *(*x + 2) + 5
3. *(*(x + 1))
4. *(*(x) + 2) + 1
5. * (*(x + 1) + 3)
6. *n
7. *(n +2)
8. (*(n + 3) + 1
9. *(n + 5)+1
10. ++*n

a. 9
b. 13
c. 4
d. 3
e. 2
f. 12
g. 14
h. 7
i. 1
j. 8
k. 5
l. 10
m. 6

Match the following with reference to the following program segment:

```
unsigned int arr[ 3 ][ 3 ] = {
2, 4, 6,
9, 1, 10,
16, 64, 5
} ;
```

1. **arr
2. **arr < *(*arr + 2)
3. *(arr + 2) / (*(*arr + 1) > **arr)
4. *(arr[1] + 1) | arr[1][2]
5. *(arr[0]) | *(arr[2])
6. arr[1][1] < arr[0][1]
7. arr[2][[1] & arr[2][0]
8. arr[2][2] | arr[0][1]
9. arr[0][1] ^ arr[0][2]
10. ++**arr + --arr[1][1]

a. 64
b. 18
c. 6
d. 3
e. 0
f. 16
g. 1
h. 11
i. 20
j. 2
k. 5
l. 4