What will be displayed in the screen after the following program executes, provided the formal parameter denoted with & will be transferred by:

a. value
b.value/result
c. reference
d. name

```
1.  int a[5];
2.  int i,X=1;

3.  void swap (int&a, int&b,int&c)
4.  int t;
5.  {
6.  c++;
7.  t = a+X+c;
8.  a = b;
9.  b = t;
10. }

11. void main()
12. {
13.       for (i=0;i<5;i++) a[i] = 6 – i;
14.       i = 2;
15.       swap(i, a[i],X);
16.       printf("%d%d%d%d%d%d",i,a[0], a[1], a[2], a[3], a[4]);
17. }
```

Solution
a. By value

Passing by value is to copy value of outside variables from function call's argument list to the inside variables of parameter in function.

At step 11, no value are returned, and no variables in function call's argument list are modified

		Global scope							Swap function			
Step	Line	i	a[0]	a[1]	a[2]	a[3]	a[4]	X	a	b	c	t
1	2							**1**				
2	10							1				
3	13	**5**	**6**	**5**	**4**	**3**	**2**	1				
4	14	**2**	6	5	4	3	2	1				
5	15	2	6	5	4	3	2	1				
6	5	2	6	5	4	3	2	1	**2**	**4**	**1**	
7	6							1	2	4	**2**	
8	7							1	2	4	2	**5**

9	8							1	**4**	4	2	5
10	9							1	4	**5**	2	5
11	10							1				
12	15	2	6	5	4	3	2	1				
Print	16	2	6	5	4	3	2	1				

=> Answer: 265432

b. By value/result

Passing by value - result is to copy value of outside variables from function call's argument list to the inside variables of parameter in function, and at the end of the function, it copies the values of inside variables back to the argument list of outside variables respectively

At step 11, no value are returned. However, the swap function assigns variables in function call's argument list with parameters in function respectively.

		Global scope							Swap function			
Step	Line	i	a[0]	a[1]	a[2]	a[3]	a[4]	X	a	b	c	t
1	2							**1**				
2	10							1				
3	13	**5**	**6**	**5**	**4**	**3**	**2**	1				
4	14	**2**	6	5	4	3	2	1				
5	15	2	6	5	4	3	2	1				
6	5	2	6	5	4	3	2	1	**2**	**4**	**1**	
7	6							1	2	4	**2**	
8	7							1	2	4	2	**5**
9	8							1	**4**	4	2	5
10	9							1	4	**5**	2	5
11	10	**4**		**5**				**2**				
12	15	4	6	5	4	3	2	2				
Print	16	4	6	5	4	3	2	2				

=> Answer: 465432

c. By reference
Passing by reference is to copy the address of the variables in function call's argument list to the inside variables of parameter in function. Therefore, the variable inside the function is a representation of the variable in function call.

At step 6, function copies the address of variables in function call's argument list to the prototype respectively.
Hence, in step 9 and 10, variables i and a[4] are modified respectively.

		Global scope							Swap function			
Step	Line	i	a[0]	a[1]	a[2]	a[3]	a[4]	X	a	b	c	t
1	2							**1**				
2	10							1				
3	13	**5**	**6**	**5**	**4**	**3**	**2**	1				
4	14	**2**	6	5	4	3	2	1				
5	15	2	6	5	4	3	2	1				
6	5	2	6	5	4	3	2	1	i	a[i]	X	
7	6							**2**	i	a[i]	X	
8	7							2	i	a[i]	X	**6**
9	8	**4**						2	i	a[i]	X	6
10	9	4					**6**	2	i	a[i]	X	6
11	10	4					6	2	i	a[i]	X	6
12	15	4	6	5	4	3	6	2				
Print	16	4	6	5	4	3	6	2				

=> Answer: 465436

D. By name

Passing by name means all variables in the function can be replaced by the name of variables in function call's argument list. Passing by name does not modify or cause any effects to the values of function call's argument list.

With the function call "swap(i, a[i],X)", the swap function can be rewrited as

```
3.void swap ()
4.int t;
5.{
6.X++;
7.t = i+X+X;
8.i = a[i];
9.a[i] = t;
10.}
```

Although passing by name does not modify or cause any effects to the values of function call's argument list, variable i, X and array a[] are declared in the global scope, so any changes of these variables in swap function affect variables in global scope.

		Global scope							Swap function			
Step	Line	i	a[0]	a[1]	a[2]	a[3]	a[4]	X	a	b	c	t
1	2							**1**				
2	10							1				
3	13	**5**	**6**	**5**	**4**	**3**	**2**	1				
4	14	**2**	6	5	4	3	2	1				

5	15	2	6	5	4	3	2	1				
6	5							1	‘i’	‘a[i]’	‘X’	
7	6							**2**	2	4	**2**	
8	7							2	2	4	2	**6**
9	8	**4**						2	**4**	4	2	6
10	9	4					**6**	2	4	**6**	2	6
11	10	4					6	2				
12	15	4	6	5	4	3	6	2				
Print	16	4	6	5	4	3	6	2				

=> Answer: 465436