CSE/IT Batch: Hinglish

Subject: Programming in C

Arrays and Pointers

DPP-01

[MSQ]

1. Which of the following declarations are INVALID?

(a) int b[][4];
(b) int b[];
(c) int b[2][][2]={1,2,3,4};
(d) int b[][2][2]={1,2,3,4};

[MCQ]

2. Consider the following two statements:
P: int a[3]={1, 2, 3};
printf("%d", *a++);

Q: int a[3]={1, 2, 3};
int *p=a;
printf("%d", *p++);
Which of the following statements is/are CORRECT?

(a) P only. (b) Q only.
(c) Both P and Q. (d) Neither P nor Q.

[MCQ]

3. Consider the following program:

```
#include<stdio.h>
int main(void)
 {
     int a[5]={5, 10, 15};
     printf("%d", 1[a]);
     return 0;
 }
```

The output is-

(a) 5 (b) 10
(c) Garbage value (d) Compilation error

[MCQ]

4. Consider the following program:

```
#include<stdio.h>
int main(void)
 {
     int 5[a]={5, 10, 15};
     printf("%d", 1[a]);
     return 0;
 }
```

The output is-

(a) 5 (b) 10
(c) Garbage value (d) Compilation error

[MCQ]

5. Consider the following program:

```
#include<stdio.h>
int main(void) {
    int a[5]={5, 10, 15, 20, 25};
    printf("%u", a);
    printf("%u", *(a+3));
    printf("%u", a+2);
    printf("%u", *(a+2)+6);
    printf("%u",*(a+*(a+1)-6));
    return 0;
 }
```

Assuming the base address of the array to be 1000 and integer size as two bytes the output is-

(a) 1000 20 1004 21 25

(b) 5 20 15 21 25

(c) 1000 20 1002 21 24

(d) Compilation error

[MCQ]

6. Consider the following program:

```
#include<stdio.h>
int main(void)
 {
     int a[5]={5, 10, 15, 20, 25};
     printf("%u\t", *(1+a));
     printf("%u\t", &a+1);
     return 0;
 }
```

Assuming the base address of the array to be 1000 and integer size as four bytes the output is-

(a) 1004 1020 (b) 10 1016
(c) 10 1020 (d) 1004 1016

Answer Key

1. **(a, b, c)**
2. **(b)**
3. **(b)**
4. **(d)**
5. **(a)**
6. **(c)**

Hints and solutions

1. (a, b, c)

(a) int b[][4]: Invalid as elements are not specified.

(b) int b[]; Invalid as size is not specified.

(c) int b[2][][2]={1,2,3,4}; Invalid. If the elements are specified, only first dimension can be omitted.

(d) int b[][2][2]={1,2,3,4}; Valid. If the elements are specified, only first dimension can be omitted.

2. (b)

int a[3]={1, 2, 3};

Array name without subscript denotes the base address of the array. So, a++ is not allowed. Hence, P is incorrect.

Q is correct.

3. (b)

The printf() statement can be interpreted as-

printf("%d", 1[a]) is equivalent to printf("%d", *(1+a));

1000	1002	1004
5	10	15

So, *(1+a) is equivalent to *(1+1000). Here, 1 signifies the increment by 1*2 bytes= 2 bytes.
So, *(1002) is 10.

4. (d)

int 5[a]={5, 10, 15}; // It is an invalid declaration.

So, compilation error will happen.

5. (a)

1000	1002	1004	1006	1008
5	10	15	20	25

printf("%u", a);//1000
printf("%u", *(a+3));//*(1000+2*3) i.e *1006 i.e 20
printf("%u", a+2);//1000+2*2=1004
printf("%u", *(a+2)+6);//*(1000+2*2)+6 i.e *1004+6 i.e 15+6 i.e 21
printf("%u",*(a+*(a+1)-6));

//*(a+*(1000+2*1)-6) = *(a+4) = *(1000+2*4) = *1008 i.e 25
Output: 1000 20 1004 21 25

6. (c)

1000	1004	1008	1012	1016
5	10	15	20	25

printf("%u\t", *(1+a));//*(1*4+1000)=*1004=10
&a+1 signifies the next 1D array. So, size incrementing by 1 means increase by 4*5 bytes.
printf("%u\t", &a+1);//1000+20=1020 is printed.
Output: 10 1020