

Array of Pointers - 2

The declaration

```
int (*p) [5];
```

means

Choose any one

- ☐ p is one dimensional array of size 5, of pointers to integers.
- ☒ p is a pointer to a 5 elements integer array. ✓ correct answer
- ☐ The same as int *p[5];
- ☐ None of these.

Array of Pointers - 3

What would be the output for the following program?

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

Choose any one

- ☐ 2 2
- ☐ 1 3
- ☒ 2 3 ✓ correct answer
- ☐ 1 2

Arrays: Pointers - 1

What would be the output of the below program

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

Choose any one

- ☐ 0
- ☐ 2
- ☐ Garbage Value
- ☒ Syntax Error ✓ correct answer

Arrays: Pointers - 2

What would be the output of the below program

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

Choose any one

- ☐ 11
- ☒ 12 ✓ correct answer
- ☐ Garbage Values
- ☐ Syntax Error

Pointer to Pointer - 1

Consider the following declaration

```
int a, *b = &a, **c = &b;
```

The following program fragment

```
a = 4;  
**c = 5;
```

Choose any one

- ☐ does not change the value of a
- ☐ assigns address of c to a
- ☐ assigns the value of b to a
- ☒ assigns 5 to a ✓ correct answer

Pointer to Pointer - 2

```
#include<stdio.h>

int main()
{
    int ***r, **q, *p, i=8;
    // address of i is 1000, p is 1200, q is 1400, r is 1600
    p = &i;
    q = &p;
    r = &q;
    printf("%d %d %d", *p, **q, ***r);
    return 0;
}
```

Your answers

1.) Write the output of the program If the integer is 4 bytes long?

8 8 8

