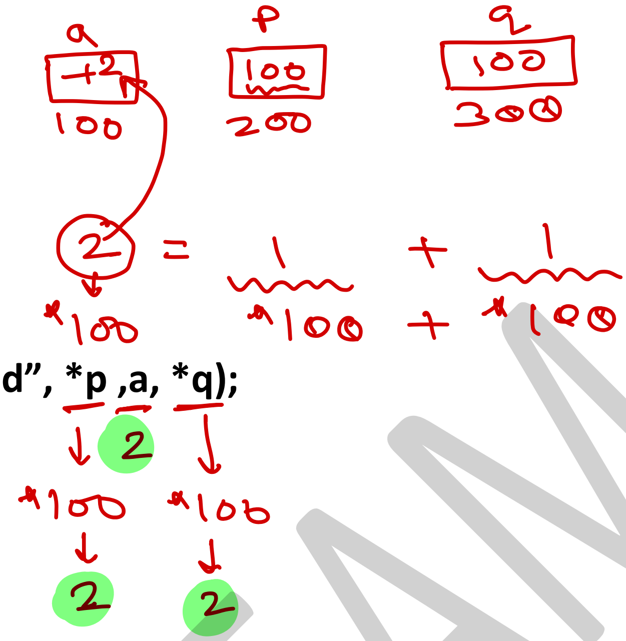
1. #include <stdio.h>

```
int main(void)
{
        int a = 1;
        int *p = &a;
        int *q = p;
        *p = *p + *q;
        printf("%d%d%d", *p ,a, *q);
        return 0;
}
```



A. 111
B. 222
C. 211
D. 221

Answer: B

2. #include <stdio.h>

```
void callbyAddress1(int *x)
{
        x=x+10;
}

void callbyAddress2(int *x)
{
        *x=*x+10;
}
int main(void)
{
        int a=10;
        printf(" %d ",a);
        callbyAddress1(&a);
        printf(" %d ",a);
        callbyAddress2(&a);
        printf(" %d ",a);
        callbyAddress1(&a);
        printf(" %d ",a);
        callbyAddress2(&a);
        printf(" %d ",a);
        return 0;
}
```

10

10

20

20

30

x

~~1040 1000 2000~~

~~1000 + 10 * 4 = 1040~~ s.f. of int*

~~destroyed when fn returns.~~

x

~~1000 3000~~

*x = *1000 = 10

*x + 10 = 20

10

*x = *1000

a

10 20

1000

x ptr is modified and var value

*x is incr by 10

20 + 10 = 30

a

✓A. 10 10 20 20 30

B. 10 10 10 10 10

C. 10 20 30 40 50

D. compile time error

Answer: A

3. #include <stdio.h>

```
int main(void)
{
        int arr1[]; ✗ → arr size is must.
        int arr2[] = { 11, 22, 33 }; ✓ → arr size taken by compiler
        int arr3[3] = { }; ✓ → arr init with empty list.
                                          all eles become 0.
        int arr4[2] = { 10, 20, 30 }; ✗ → size is less.
        int arr5[6]; ✓ arr declaration - eles are garbage.
        int [] arr6; ✗ invalid syntax.
        return 0;
}
```

Which variable declarations are wrong?

A. arr1, arr3, arr4, arr6

B. arr1, arr3, arr6

✓C. arr1, arr4, arr6

D. None of these

Answer: C

```
int n = 10;        ↗ error : C89  ✗
int arr[n];        ↘ allowed : C99 on gcc. ✓
```

4.
```
#include <stdio.h>

int main(void)
{
        double arr[3];
        char *p1, **p2;

        printf("%u, ", sizeof(arr) + sizeof(arr[0]));

        printf("%u\n", sizeof(*p1) + sizeof(*p2) + sizeof(p1) +
sizeof(p2));

        return 0;
}
```

sizeof (**p2)

p2 → 8 bytes

p2 → char → 8 bytes

*(*p2) → char → 1 byte

double

0th ele

8*3 + 8 = 32

size of whole array

size of one ele in array

char

char*

1 + 8 + 8 + 8

Note: consider 64-bit compiler.

✓ A. 32, 25
B. 32, 28
C. 64, 25
D. None of these

Answer: A