

1. int main(void)

```
{
```

```
    int a = 10;
```

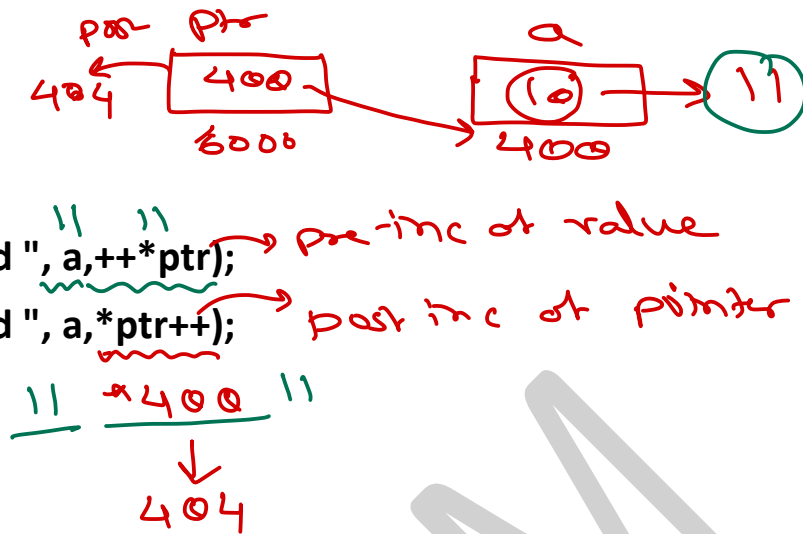
```
    int *ptr = &a;
```

```
    printf(" %d %d ", a, ++*ptr);
```

```
    printf(" %d %d ", a, *ptr++);
```

```
    return 0;
```

```
}
```



✓ A. 11 11 11 11

B. 11 11 12 12

C. Compile time error

D. Run time error

Answer: A

C fn args are processed right to left.

③ ② ①
fn (a++, b++, c++);



2. #include<stdio.h>

```
int print_size(int a[])  
{  
    printf("%d, %d, ", sizeof(a), sizeof(a[3]));  
    return 0;  
}
```

Handwritten notes for `print_size`:
- `int a[]`: *formal arg → pointer.*
- `int a`: *Compiler convert*
- `sizeof(a)`: *size of pointer on 32 bit. → 4*
- `sizeof(a[3])`: *array ele size = 4*

```
int main(void)  
{  
    int a[] = {1,2,3};  
    printf("%d, %d, ", sizeof(a), sizeof(a[-1]));  
    print_size(a);  
}
```

Handwritten notes for `main`:
- `int a[] = {1,2,3};`: *array of 3 int size = 3 × 4 = 12 bytes.*
- `sizeof(a)`: *12*
- `sizeof(a[-1])`: *index array element → 4*
- `print_size(a)`: *by addr. → 4*

A. compile time error

B. 12 , 4 , 12 , 4

✓ C. 12 , 4 , 4 , 4

D. 12 , 2 , 12 , 8

Answer: C



3. #include<stdio.h>

int main(void)

{

int a = 4;

int * ptr = &a;

*ptr = update(a);

printf("a = %d ptr = %d ", a, *ptr);

printf("a = %d ptr = %d ", a, ++*ptr);

printf("a = %d ptr = %d ", a, --*ptr);

return 0;

}

int update(int a)

{

int value=(a+a*a+a);

return value;

}

A. a = 23 ptr = 23 a=24 ptr =24 a=23 ptr =23

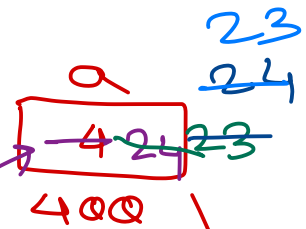
B. a = 63 ptr = 63 a = 64 ptr = 64 a = 63 ptr = 63

C. a = 24 ptr = 24 a = 25 ptr = 25 a = 24 ptr = 24

D. a = 64 ptr = 64 a = 65 ptr = 65 a = 64 ptr = 64

E. Compile time error

Answer: A



*ptr = 24;

23

24

23

24

23

24

23

pre-dec val.

23

pre-inc val.

24

pre-dec val.

23

copy

4+4*4+4



24



4. #include<stdio.h>

int main(void)

{

char s[]={'a','b','c','\n','c','\0'};

char *p,*str,*str1;

p=&s[3];

str=p;

str1=s;

printf("%d",++*p++*str1-32);

return 0;

}

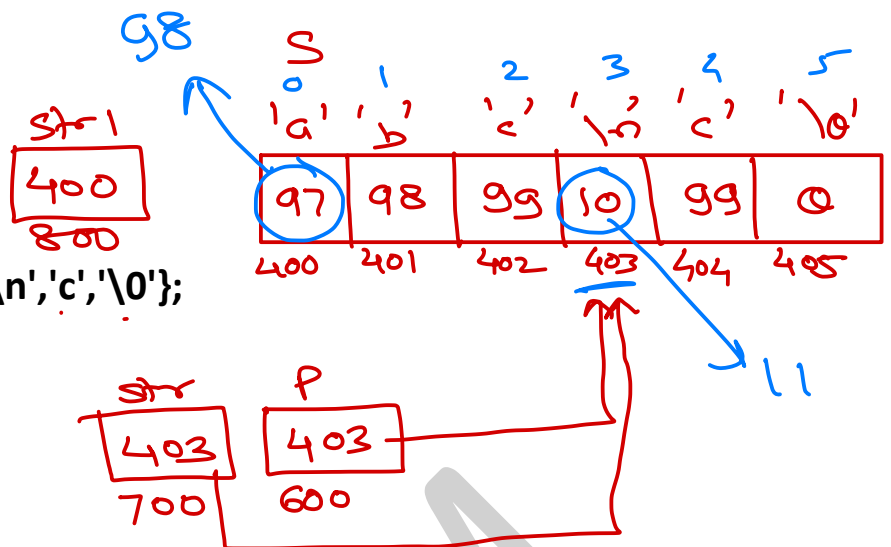
A. 77

B. 88

C. 76

D. 75

Answer: A



pre inc value pre inc value

$$11 + 98 - 32 = 77$$

5. #include<stdio.h>

int main(void)

{

char a[100]; *mm*

a[0]='a';a[1]='b';a[2]='c';a[4]='d';

abc(a);

return 0;

}

abc(char a[])

{

address

a++; *b* printf("%c",*a);

a++; *c* printf("%c",*a);

}

A. ac

✓ B. bc

C. cc

D. cd

Answer: B

0	1	2	3	4	5	...	99
'a'	'b'	'c'	'x'	'd'	'x'	'x' - 'x' - 'x'	'x'
97	98	99		100			
100	101	102	103	104			

