

C & DSA

1 →

`int *f(int *);`

B

f is a fⁿ which takes int pointer as an argument and returns address of an integer

Ans

2 →

`main()`
{

A

`extern int i;`

`i = 20;`

`printf("x.d", i);`

}

→ Linking Error

3 →

`void main()`
{

4 Ans

`static int i = 5;`

`if (--i)`

{ `main();`

`printf("x.d", i);`

}

}

↳ How many Zero's = 4

4

A

```
main()
{
    void prn();
    prn();
    prn();
    prn();
}
```

→ 66 →
→ 67 →
→ 68 →

char
B
C
D

```
void prn()
{
    static int i = 1;
    printf("%c", (65 + i++));
}
```

→ BCD Ans

5

A

```
int k = (i < 3) + (j >> 2)
```

→ $\text{int } k = i * 8 + j / 4;$

6

A

```
int foo(unsigned int n)
{
    int c, x = 0;
    while (n != 0)
    {
        if (n & 01) x++;
        n >>= 1;
    }
    return c;
}
```

→ It counts the total no. of bits set in an unsigned integer

7

```
int f (int a, int b)
{
```

```
    if (b == 0)
```

```
        return 1;
```

```
    else if (b % 2 == 0)
```

```
        return (f(a, b/2) * f(a, b/2));
```

```
    else
```

```
        return (a * f(a, b/2) * f(a, b/2));
```

```
}
```

↙

$$\begin{aligned} f(2, 10) &= f(2, 5) * f(2, 5) \\ &= (2 * f(2, 2) * f(2, 2))^2 \\ &= (2 * f(2, 1)^4)^2 \\ &= 4 * (2 * f(2, 0) * f(2, 0))^8 \\ &= 2^2 * 2^8 * 1 * 1 \\ &= 2^{10} \text{ Ans} \end{aligned}$$

8

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

```
#define R 10
```

```
#define C 20
```

```
int main()
```

```
{ int (*p)[R][C];
```

```
    printf("%d", size of (*p));
```

```
    getch();
```

```
    return 0;
```

```
}
```

$(10 \times 20) \times 4$

$= 800 \text{ Ans}$

size of (*p) = 800

Ans

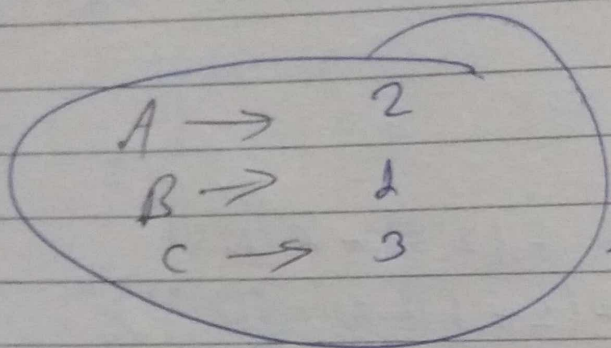
9

I

- (A) typedef int(*ptrn)(); ptrn p;
- (B) int(*p)[4];
- (C) int *p[4];

- (1) pointer to an array of int
- (2) pointer to fⁿ returning int
- (3) Array of pointers pointing to

B



Ans

10

D

```
int i;  
main()  
{ i=3; S(); R(); }  
void S()  
{ print(i);  
  print(" ");  
}  
void R()  
{  
  int i=2;  
  S();  
}
```

Static	Dynamic
3 3	3 2

Ans