

Quiz

Date... 8/10/23

① → Method used to count prime No

② → All of these → Naive Approach
Sqrt Approach
Sieve of Eratosthenes

③ → why 1 is not considered as Prime?
A) 1 has only one factor

④ → what is Complexity calculating that if a no is prime or not in b/w 1 & n-1?

(A) $O(n)$

⑤ → TC for sqrt approach prime No?
(B) $O(n \cdot \sqrt{n})$

⑥ → which of the following is the correct implementation for isPrime() in naive approach?

```
bool isPrime(int n) {  
    if (n == 1) return false;  
    for (int i = 2; i < n; i++)  
        if (n % i == 0) return false;  
    return true;  
}
```

⑦ → TC → Sieve of Eratosthenes?
C) → $O(n \log(\log n))$

Date 8/10/23

(7) → TC → optimise sieve of eratosthenes?

(a) → $O(n \log(\log n))$

(8) → first multiple in segmented sieve

(c) → first multiple = $(l/p) * p$
if (first multiple < l)

first multiple += p

(9) → Improvement in segmented sieve

(1) → for (auto p : base primes)

(2) → int fm = $(l/p) * p$;

(3) → if (fm < l) fm += p;

(4) → int j = fm;

(5) → while (j ≤ r)

(6) → seg sieve[j] = false; → seg sieve[j - l]

(7) → j = j + p;

(8) → y

(9) → y

(c) → (1) & (6)

(10) → which of the following is done for optimisation in sieve of eratosthenes?

(c) → 1. Inner loop

↳ $j = i * i$

outer loop end

(or) $i * i = \text{sqrt}(n)$
 $i * i = n$

Date 9/10/23...

(11) \rightarrow Optimize code \rightarrow worst TC $\rightarrow a^b$?
(2) $\rightarrow O(\log b)$

(12) \rightarrow pseudo code for calculating exponents

(2) \rightarrow

```
int ans = 1;
while (b > 0) {
    if (b & 1) {
        ans = ans * a;
    }
    a = a * a;
    b >>= 1;
}
```

(13) \rightarrow Evaluate the expression 6359, 320

$\text{if } (\text{gcd}(a-b, b)) \mid a > b$
 $a \rightarrow 6359, b \rightarrow 320$
 $\text{else } (b > a)$
 $\text{gcd}(b-a, b)$

(2) $\rightarrow 279$

(14) $\rightarrow (12 * 13) \% 5$
 $((12 \% 5) * (13 \% 5)) \% 5$
 $(7 * 8) \% 5$
 $56 \% 5 = 1 \leftarrow \text{A}$

(15) \rightarrow Incorrect for gcd

(2) \rightarrow None

Correct $\rightarrow \text{gcd}(a, b) = \text{gcd}(a \% b, b)$
 $\text{gcd}(a, b) = \text{gcd}(b, a \% b)$

Date... 9/10/23

$\text{gcd}(a, b) = \text{gcd}(a-b, b), a > b$
 $\text{gcd}(b-a, a), b > a$

(16) → The difference operator is → (A) *

(17) → String * x, y

(A) → x → Pointer, y → String

(18) → Which of following is wrong?

(A) → $\text{int} * p = 0$

(B) → $\text{string } s, *sp = 0;$

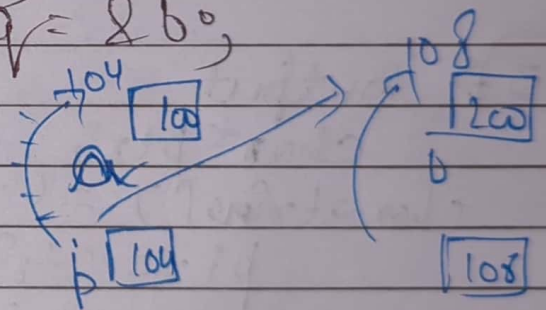
(C) → $\text{int } i; \text{double } *dp = 2i;$

(D) → None

double pointer address int
 → Wrong

(19) → $\text{int } a = 100, b = 200;$
 $\text{int } *p = \&a, \text{int } *q = \&b;$
 $p = q$

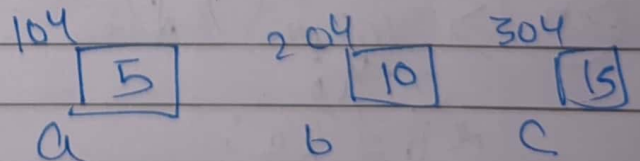
(B) → p now points to b



$p = q$

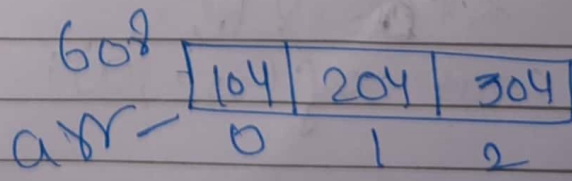
(20) → $a = 5, b = 10, c = 15$

$\text{int } arr[3] = \{ \&a, \&b, \&c \}$
 $arr[1]$



$arr[1] \rightarrow 204$

(B) → Address value of some element



Date... 9/30/23...

(21) → function → pointer to float
 → pointer to a pointer to a char

→ returns a pointer to a pointer to a pointer to

② → int^{**} function(float^* , char^{**})

(22) → $\text{char arr}[20];$
 $\text{int } i;$
 $\text{for}(i=0; i<10; i++)$
 $\quad *(&\text{arr} + i) = \text{G} + i;$

$*(&\text{arr} + i) = \text{G} + i;$

$\text{cout} << \text{arr};$

A B C D E F G H I J
 G S + 0 1 2 3 4 5 6 7 8 9

① → A B C D E F G H I J

(23) → Output

$\text{char}^* \text{ptr};$ 0 1 2 3 4 5
 $\text{char str}[7] = \text{"abcdefg"};$
 $\text{ptr} = \text{str};$
 $\text{ptr} += 5;$
 $\text{cout} << \text{ptr};$

① → fg

(24) → main() d
 $a = 4;$
 $*b = 8a;$
 $**a = 8b;$
 $\text{cout} << f(c, b, a);$

$f(x, *py, **pz)$
 $**pz + 1 \rightarrow 5$
 $z = **pz \rightarrow 5$
 $*py + 2 \rightarrow 7$
 $y = *py \rightarrow 7$
 $x + 3 \rightarrow 10$

Date... 9/30/23

8 7 5
return x+y+z = 19 ← (A)

x → pass by value
change in c → Not affect x

(25) → i = 8
*p = &i → 10
(*p)++ → 9
q = &p;
**q++ → 10
r = &q

(26) → i = 8
*p = &i → 10
**q → 10
**r → 10

(26) → x = 5
p(&x);
x → 6
p(Print *y)
x = *y + 2 → 7
Print *y
*y = x - 1 → 6
cout << x → 6
z = x → 6
Print z
z + 7 = 13
cout << z → 14

(27) → char s[7] = "ABCDEF";
for (i = 0; s[i] != '\0'; i++)
cout << s[i] << *(&s[i] + 1) << *(&s[i] + 2) << endl;
s[i] = i[s];
A 65 AA
B 66 BB
C 67 CC
D 68 DD

(28) → i = 0
m = 0
x = 0
APCOM → 0 68 DD

Teacher's Sign.....

Date... 9/10/23

(28) $\rightarrow p = 5$
 $\text{int const } q = 2p$
 $p++;$ $\rightarrow 6$

Const \rightarrow value of variable cannot be changed

(B) $\rightarrow 6$

(29) $\rightarrow \text{int const } p = 5;$
 $++p$

(D) \rightarrow Error \rightarrow because value of const cannot be changed

(30) \rightarrow How to Dynamically create a 2-D Array in C++

(B) $\rightarrow \text{int}^{**} \text{arr} = \text{new int}^{*}[n];$
 $\text{for } () \text{ of } \text{arr}[i] = \text{new}[n];$

