

Q1 A language translator converts a high-level source code to machine level instructions.

The two types of language translators for high level languages are: Compiler and interpreter

COMPILER

- Translates the entire source code to machine instructions at a time.
- Faster ~~program~~ compared to an interpreter.
- Best for debugging since all the errors are conveyed to the programmer at once.
- For large codes, it becomes more difficult to implement fixes and changes for the developer when the compile time increases significantly.

Eg → C

INTERPRETER

- Translates each sentence, one by one to machine language instructions.
- Slower process.
- Since the instructions get translated line-by-line, really it isn't that good for debugging, since only one problem will be highlighted.
- Easier and much faster than compilation to implement fixes and changes.

Eg → Python

Q2 a) #0 #1 #2 #3 #4 #5 #6 #####

b) given $i=1, j=2, K=3$ and $m=4$

$$j * k = m + \frac{m}{j} * k$$

$$k = m + \frac{m}{j} * k = 4 + \frac{4}{2} * 3 = 4 + 2 * 3$$

$$\boxed{k=6}$$

$$j * k \Rightarrow j = 2 * 6 = \boxed{12}$$

c) Output:
bbb

Since the first if expression evaluates to zero, it is skipped.

case 2:

Output:

aaa

bbb

If is skipped only when the expression equates to zero

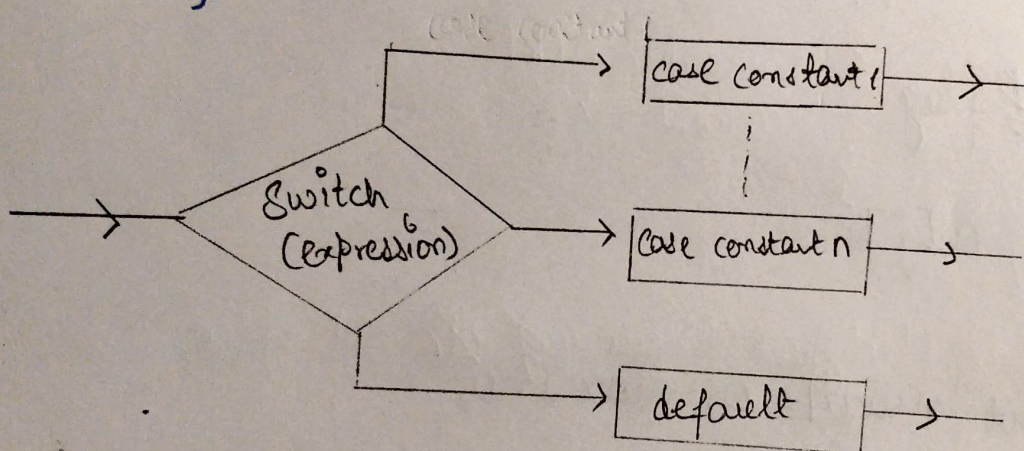
Q2 d) switch (expression) {

case constant₁: expression;
break;

case constant_n: expression;
break;

default: expression;
break;

}



Q3

$B_1 = a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z$

$B_2 = s[i] \quad i = '10'$

$B_3 = \text{flag}$

Q3
B₄ =
B₅ ⇒

Q3 B₁ ⇒ ~~a~~, b, c, d, e, f, g, h, i, j, k, l,
m, n, o, p, q, r, s, t, u, v, w, x,

B₂ ⇒ s[i] != 'o' y, z

B₃ ⇒ flag

B₄ ⇒ a[j]

B₅ ⇒ int maxM = 0, maxI = 0;

for (i = 0; i < k; i++)

```
{  
    if (maxM < a[i])  
    {  
        maxM = a[i];  
        maxI = i;  
    }  
}
```

B₆ ⇒ printf("Highest frequency %d of %c", maxM, c[maxI]);