

Enroll no. - NBTG13715

Name - Nitin chandhary

Batch - FB

Tut - 9

Date 12-12-2021
Page

Ans ① →

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

```
void main() {
```

```
    char str[10];
```

```
    char ch;
```

```
    printf("\n Enter the string :");
```

```
    gets(str);
```

```
    printf("\n Enter character which you want to delete:");
```

```
    scanf("%c", &ch);
```

```
    del(str, ch);
```

```
}
```

```
void del(char str[], char ch) {
```

```
    int i, j = 0;
```

```
    int size;
```

```
    char chl, *pb = &ch;
```

```
    char strl[10];
```

```
    size = strlen(str);
```

```
    for(i = 0; i < size; i++) {
```

```
        if(*str+i != *pb) {
```

```
            chl = *str+i;
```

```
            *strl+j = chl;
```

```
            j++;
```

```
        }
```

```
    }
```

```
    strl[j] = '\0';
```

```
    printf("Corrected string is : %s", strl);
```

Output →

Enter the string: hitin

Enter character which you want to delete: n

Corrected string is: iti

Ans (2) →

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

```
int main()
```

```
{
```

```
int n, array[100], i;
```

```
printf("enter how many numbers you want to enter");
```

```
scanf("%d", &n);
```

```
printf("enter numbers\n");
```

```
for (i=0; i<n; i++)
```

```
{
```

```
    scanf("%d", &array[i]);
```

```
}
```

```
int lcm = array[0];
```

```
int gcd = array[0];
```

```
for (int i=1; i<n; i++) {
```

```
    gcd = findgcd(array[i], lcm);
```

```
    lcm = (lcm * array[i]) / gcd;
```

```
}
```

```
printf("LCM: %d", lcm);
```

```
}
```

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

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

```
        return a;
```

```
    return findgcd(b, a % b);
```

```
}
```


Output -

enter how many numbers you want to enter 7
enter numbers

1

2

3

4

5

6

7

LCM: 420

Ans ③ ① → 30

② Calculation will be done in this way →

$$\begin{aligned} & i * j * i + * j \\ & = 3 * 3 * 3 + 3 \\ & = 27 + 3 = 30 \end{aligned}$$

④ $x=31, y=6922032, z=6922032$

printed by printf as both are
integer addresses of x

⑤ Hello

This is due to the fact that name of
array points towards first element's base
address.

⑥ 8, 8, 8

As all pointers pointing towards 8

⑦ K

As str will be a null string

⑧ 8, 1

⑨ JACK

⑩ 8, 10

⑪ 1006, 2, 2

↳ Address of second element

(J) 2, 0

(K) 300

(L) ce
as pointer will reach to c

(M) hello

(N) hello

(O) BndiaBIIdiaBIXia

(P) Alice like ice ce e