

* prog 1 :- prime factors of a numbers.

$$N = 60$$

$$2, 3, 5$$

$$\begin{array}{r} 2 \mid 60 \\ 2 \mid 30 \\ 3 \mid 15 \\ 5 \mid 5 \\ 1 \end{array}$$

$$N = 35$$

$$5, 7$$

$$\begin{array}{r} 5 \mid 35 \\ 7 \mid 7 \\ 1 \end{array}$$

```
#include <iostream>
using namespace std;
int main () {
```

```
    int n; cout << "Enter a number:- " ; cin >> n;
    for (int i = 2; i <= n; i++) {
        while (n % i == 0)
```

}

$$n = n / i;$$

```
    cout << i << " ";
```

}

```
if (n != 1)
```

```
    cout << n;
```

}

```
return 0;
```

}

⇒ output :-

Enter a number:- 45
3 3 5

$$\begin{array}{r} 15 \\ 3 \overline{) 45} \\ -3 \downarrow \\ 15 \end{array}$$

⇒ Dry Run :-

$$n = 45$$

$$i = 2$$

$$i < 45 \Rightarrow 2 < 45 \Rightarrow \text{True}$$

$$n \% i == 0 \Rightarrow 45 \% 2 == 0 \Rightarrow \text{False}$$

$$i = 3$$

$$3 < 45 \Rightarrow \text{True}$$

$$45 \% 3 == 0 \Rightarrow \text{True}$$

$$n = 45 / 3 \Rightarrow 15.$$

print 3

$$i = 3$$

$$15 \% 3 == 0 \Rightarrow \text{True}$$

$$n = 15 / 3 \Rightarrow 5$$

print 3

$$i = 3$$

$$5 \% 3 == 0 \Rightarrow \text{False}$$

↳ exist while loop &
got for loop

$$i = 4$$

5 < 4 ~~==~~ false

$$i < 5 \Rightarrow 4 < 5 \Rightarrow \text{True}$$

$$5 \% 4 == 0 \Rightarrow \text{False}.$$

$$i = 5$$

i < 5 \Rightarrow False \Rightarrow exist for loop

if ($5! = 1$) \rightarrow true
print 5

Prime factorisation of 45 completed.

→ Greeks for Greeks :- Problem of the Day :- 09/08/23
prog 2 :- Largest prime factor of a number.

→ $n = 24$

$$\begin{array}{c|c} 2 & 24 \\ \hline 2 & 12 \\ \hline 2 & 6 \\ \hline 3 & 3 \\ \hline & 1 \end{array}$$

$$\text{Ans} \Rightarrow \underline{\underline{3}}$$

→ 36

1, 2, 3, 4, 6, 9, 12, 18, 36

i i=2

$$\left| \begin{array}{l} \frac{n}{i} \quad \frac{n}{i} \\ \hline \end{array} \right.$$

$$i^2 \leq \frac{n}{i}$$

$$i^2 \leq n \Rightarrow i \leq \sqrt{n}$$

```
#include <iostream>
#include <cmath>
using namespace std;
int main () {
    int n;
    cout << "Enter Number:- ";
    cin >> n;
```

```
long long largest_prime_factor = 0;
while (n % 2 == 0) {
    largest_prime_factor = 2;
    n = n / 2;
```

```
} for (int i = 3; i <= sqrt(n); i = i + 2)
```

```
while (n % i == 0) {
    largest_prime_factor = i;
    n = n / i;
```

```
} if (n > 2) {
```

```
    largest_prime_factor = n;
```

```
cout << "Largest prime factor :- " <<
        largest_prime_factor;
```

```
return 0;
```

```
}
```

⇒ output :-

Enter number:- 45

Largest Prime Factor :- 5

⇒ Dry Run

$$n = 45$$

largest prime number = 0

while ($45 \% 2 == 0$) \Rightarrow false

for (i = 3; i < $\sqrt{45}$; i = i + 2) {

while ($45 \% 3 == 0$) \Rightarrow true.

$$\begin{array}{r} 3 \mid 45 \\ 15 \end{array}$$

largest prime number = 3

$$n = 45 / 3 = 15$$

↳ while loop exist.

while loop enter again

while ($15 \% 3 == 0$) \Rightarrow true.

largest prime number = 3

$$n = 15 / 3 = 5$$

↳ while loop exists

⇒ for loop exist

for (i = 5, i < $\sqrt{5}$; i = i + 2)

$5 \Rightarrow 2 \cdot 2 \cdot 3 \Rightarrow$ not possible.

for loop permanently exist.

if ($5 > 2$)

\Rightarrow print 5

* prog 3 :- Smallest prime factor of a number.

$$n = 24$$

(2) Ans

$$\begin{array}{r} 2 | 24 \\ 2 | 12 \\ 2 | 6 \\ 3 | 3 \\ \hline & 1 \end{array}$$

$$n = 49$$

(7) Ans

$$\begin{array}{r} 7 | 49 \\ 7 | 7 \\ \hline & 1 \end{array}$$

```
#include <iostream>
#include <cmath>
int smallest_primefactor ( int n ) {
    if ( n <= 1 )
        return -1; // Invalid input
    else if ( n % 2 == 0 )
        return 2;
    else
        for ( int i = 3 ; i <= sqrt ( n ) ; i = i + 2 )
            if ( n % i == 0 )
                return i;
```

```
        return n;
    }

int main()
{
    int num;
    cout << "Enter number :- ";
    cin >> num;
    cout << "Smallest Prime Factor :- ";
    cout << smallest_prime_factor(num);
    return 0;
}
```

3 output :-
Enter number :- 45
Smallest prime factor :- 3

* Dry run :-

```
num = 45
if (45 <= 1)    → false
else if (45 % 2 == 0) → false
else if
```

```
{   for (i=3; i <= sqrt √45; i=i+2)
    {
        if (45 % 3 == 0)
    }
    return 3
}
```

Point 3

* prog 4 :- Reverse a number

Num \rightarrow 1 2 3 4

Reverse \rightarrow 4 3 2 1

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int n;
```

```
    cout << "
```

```
Enter number:-";
```

```
    cin > n;
```

```
    int reverse = 0;
```

```
    while (n > 0) {
```

```
        int lastdigit = n % 10;
```

```
        reverse = reverse * 10 +
```

```
        lastdigit;
```

```
        n = n / 10;
```

```
}
```

```
    cout << "Reverse of number :-";
```

```
    return 0;
```

```
<< reverse;
```

```
}
```

$$\begin{array}{r} 123 \\ \times 10 \\ \hline 1230 \end{array}$$

→ output:-

Enter number :- 1234

Reverse of number :- 4321

→ Dry run :-

$n = 1234$

while ($1234 > 0$) \Rightarrow true.

$lastdigit = 1234 \% 10 = 4$

$reverse = 0 * 10 + 4 = 4$

$n = 1234 / 10 = 123$

while ($123 > 0$) } true

{

$$\text{lastdigit} = 123 \% 10 = 3$$

$$\text{reverse} = 4 * 10 + 3 = 43$$

$$n = 123 / 10 = 12$$

{

while ($12 > 0$) } true

{

$$\text{lastdigit} = 12 \% 10 = 2$$

$$\text{reverse} = 43 * 10 + 2 = 432$$

$$n = 12 / 10 = 1$$

{

while ($1 > 0$) } true

{

$$\text{lastdigit} = 1 \% 10 = 1$$

$$\text{reverse} = 432 * 10 + 1 = 4321$$

$$n = 1 / 10 = 0$$

{

while ($0 > 0$) } false

↳ loop exits

print \Rightarrow reverse

$\Rightarrow 4321$

P E S T E A

$$+ = 0 \times P S T = + 1234567$$

$$+ = 1 + 01 * 10 = 1234567$$

$$S E T = 01 * P E S T = 1$$