

\* DSA DAY 7 :- 24/07/2024

\* Prog 1 :- ~~Eg~~ Euclidean Algorithm

GCD ( 54, 888 )

$$\begin{array}{r}
 54) \overbrace{888}^{(16)} \overbrace{24}^{(2)} \overbrace{54}^{(2)} \overbrace{24}^{(2)} \\
 \underline{54} \downarrow \qquad \qquad \qquad \underline{48} \qquad \qquad \qquad \underline{24} \\
 \underline{348} \qquad \qquad \qquad \underline{6} \qquad \qquad \qquad \underline{0} \\
 \underline{324} \\
 \underline{24}
 \end{array}$$

$$\begin{aligned}
 888 &= 54(16) + 24 \\
 \rightarrow 54 &= 24(2) + 6 \\
 24 &= 6(4) + 0
 \end{aligned}$$

$\Rightarrow$  GCD ( 54, 888 )

$$\text{GCD}(54, 888) = \text{GCD}(54)$$

$$\begin{aligned}
 \text{GCD}(888, 54) &= \text{GCD}(888 - 54, 54) \\
 &= \text{GCD}(834, 54) \\
 &= \text{GCD}(780, 54) \\
 &= \text{GCD}(726, 54) \\
 &= \text{GCD}(672, 54) \\
 &= \text{GCD}(618, 54) \\
 &= \text{GCD}(564, 54) \\
 &= \text{GCD}(510, 54)
 \end{aligned}$$

456

402

348

294

240

186

132

$$\text{GCD}(78, 54)$$

$$\text{GCD}(24, 54)$$

$$\text{GCD}(54, 24)$$

$$\text{GCD}(30, 24)$$

$$\text{GCD}(6, 24)$$

$$\text{GCD}(24, 6)$$

$$\text{GCD}(18, 6)$$

$$\text{GCD}(12, 6)$$

$$\text{GCD}(6, 6)$$

$$\text{GCD}(0, 6)$$

$$\text{GCD} \Rightarrow \underline{\underline{6}}$$

GCD is last non-zero remainder.

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int n1, n2;
```

```
    cout << "Enter Number 1 :- ";
```

```
    cin >> n1;
```

```
    cout << "Enter Number 2 :- ";
```

```
    cin >> n2;
```

```
    while (n1 > n2) (n1 > 0 && n2 > 0)
```

```
{ (n1 > n2)
```

```
    if (n1 > 0 && n2 > 0)
```

```
{
```

```
    n1 = n1 % n2;
```

```
}
```

```

use
{
    n2 = n2 / n1;
}
if (n1 == 0)
{
    percent << n2;
}
else {
    cout << n2;
}
return 0;
}
    
```

Output :-

Enter Number 1 :- 50  
 Enter Number 2 :- 10  
 10

⇒ Dry Run :-

$$n_1 = 50, n_2 = 10$$

while ( $50 > 0 \& \& 10 > 0$ ) true

{

if ( $50 > 10$ ) → true.  $10 \overline{) 50}$

$$50 \div 10 = 0$$

} ; → while loop exist

if ( $n_1 == 0$ ) → true.

cout <<  $n_2 = 10$

$$\begin{array}{r} 2 \\ (4) 50 \\ - 38 \\ \hline 12 \end{array}$$

⇒ Dry run

$$n_1 = 50, n_2 = 19 \text{ same}$$

while ( $50 > 0 \& \& 19 > 0$ ) {

if ( $50 > 19$ ) → true.

$$50 \div 19 = 12$$

}

while ( $i > 0$  &  $j > 0$ )  $\Rightarrow$  true

{

if ( $i > j$ )  $\Rightarrow$  false.

else

$$n_2 = 19 \% 12 = 7$$

}

while ( $i > 0$  &  $j > 0$ )

{

if ( $i > j$ )  $\Rightarrow$  true.

$$n_1 = i \% j = 5$$

}

while ( $5 > 0$  &  $j > 0$ )  $\Rightarrow$  true.

{

if ( $5 > j$ )  $\Rightarrow$  false

else

$$n_2 = j \% 5 = 2$$

}

while ( $5 > 0$  &  $i > 0$ )  $\Rightarrow$  true

{

if ( $5 > i$ )  $\Rightarrow$  true

$$n_1 = 5 \% i = 1$$

$$1 \frac{2}{5}$$

$$\frac{2}{5} \\ 0$$

}

while ( $i > 0$  &  $j > 0$ )  $\Rightarrow$  true.

{

if ( $i > j$ )  $\Rightarrow$  false

else

$$n_2 = i \% j = 0$$

}

$$\begin{array}{r} 1 \\ 12 \overline{) 19} \\ -12 \\ \hline 7 \end{array}$$

`while ( i > 0 & 0 > 0) false`  
 $\hookrightarrow$  while loop exist.

`if ( n1 == 0 )`  $\Rightarrow$  false

`else`

`cout << n1 = [1] only`

\* prog 2 :- sum of all odd numbers.

```
#include <iostream>
using namespace std;
int main ()
{
    int n1, n2, sum=0;
    cout << "Enter Number 1:- ";
    cin >> n1;
    cout << "Enter Number 2:- ";
    cin >> n2;
    for (int i=n1; i<=n2; i++)
    {
        if (n1 % i != 0)
            sum += i;
    }
    cout << "Sum of odd number" << sum;
    return 0;
}
```

2) output

Enter Number 1 = 1

Enter Number 2 = 20

Sum of odd number = 100

\* prog 3 :- Counting Digit.

n = 1234

~~total no.~~

count = 4

{ No of digit } ,

#include <iostream>

using namespace std;

int main()

{

int num, count = 0;

cout << "Enter Number :- ";

cin >> num;

while (num > 0),

{

num /= 10;

count += 1;

}

cout << "No of Digits :- " << count

<< endl;

return 0;

}

→ output :

Enter Number :- 1234

No of Digits :- 4

\* prog 3 :- Counting Digit using log

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
    int num;
    cout << "Enter Number :- ";
    cin >> num;
    int count = (log10(num)) + 1;
```

```
cout << ("No of Digits :- ") << count;
return 0;
```

}

→ Output:-

Enter Number :- 1234

No of digits :- 4

→ Des res

$$n = 1234$$

$$\log_{10}(1234) = 3.81$$

$$count = 3.81 + 1$$

$$= 4.81$$

$$= 4$$

= int

print (4)

## \* prog 4: Palindrome

Palindrome are whose member for which reverse is exactly same as the original one.

Eg) 121  $\Rightarrow$  121

1331  $\Rightarrow$  1331

11  $\Rightarrow$  11

7  $\Rightarrow$  7

123  $\Rightarrow$  321  $\times$

```
#include <iostream>
using namespace std;
```

```
int main ()
```

```
{
```

```
    int num, lastdigit, reverse = 0;
```

```
    cout << " Enter a Number :- " ;
```

```
    cin >> num;
```

```
    int duplicate = num;
```

```
    while (num > 0)
```

```
{
```

```
        lastdigit = num % 10;
```

```
        reverse = reverse * 10 + last  
        digit;
```

```
        num = num / 10;
```

```
}
```

```
if (reverse == duplicate)
```

```
{
```

```
    cout << " duplicate is a  
    palindrome " ;
```

```
}
```

```
    else
    {
        cout << "duplicate cc cc is not a
        palindrome " ;
    }
    return 0;
}
```

3 output:-

Enter a Number :- 121  
121 is a palindrome.

\* prog 5 :- Palindrome in range.

```
#include <iostream>
using namespace std;
int main ()
{
    int ll, ul;
    cout << " Enter Lower Limit :- " ;
    cin >> ll;
    cout << " Enter Upper Limit :- " ;
    cin >> ul;
    cout << " Palindrome numbers between "
    << ll << " and " << ul <<
    " are:- " << endl;
    for (int i = ll + 1; i <= ul; i++)
    {
        int temp = i;
        int duplicate = i;
        int reverse = 0;
        int lastdigit;
```

```
while (temp > 0)
{
    lastdigit = temp % 10 ;
    reverse = reverse * 10 + lastdigit ;
    temp /= 10 ;
}

if (reverse == duplicated)
{
    count ++ ; duplicate << endl ;
}

return 0 ;
}
```

Output :-

Enter lower limit :- ~~98~~ 98

Enter upper limit :- 1102

Palindrome numbers between 98 and 112 are :-

99

101

111

Dry run :-

ll = 98

ul = 112

```
for (i = 99 ; i <= 111 ; i++)
    temp = 99
```

duplicate = 99

~~lastdigit >= 0~~ reverse = 0 .

while ( $99 > 0$ )  $\Rightarrow$  true

{

$$\text{lastdigit} = 99 \% 10 = 9$$

$$\text{reverse} = 0 * 10 + 9 = 9$$

$$\text{temp} = 99 / 10 = 9$$

}

while ( $9 > 0$ )

{

$$\text{lastdigit} = 9 \% 10 = 9$$

$$\text{reverse} = 9 * 10 + 9 = 99$$

$$\text{temp} = 9 / 10 = 0$$

}

while ( $0 > 0$ )  $\Rightarrow$  false.if ( $99 = 99$ )  $\Rightarrow$  true.

print 99

for (i = 100; i &lt;= 111; i++)

{

~~temp = duplicate = 100~~

~~reverse = 0~~

while ( $100 > 0$ )  $\Rightarrow$  true

{

$$\text{lastdigit} = 100 \% 10 = 0$$

$$\text{reverse} = 0 * 10 + 0 = 0$$

$$\text{temp} = 100 / 10 = 10$$

}

~~while ( $10 > 0$ )~~

{

$$\text{lastdigit} = 10 \% 10 = 0$$

$$\text{reverse} = 0 * 10 + 0 = 0$$

$$\text{temp} = 10 / 10 = 1$$

}

while ( $\text{temp} > 0$ )  $\Rightarrow$  true.

$$\text{last digit} = 1 \% 10 = 1$$

$$\text{reverse} = \cancel{0} * 10 + 1 = 1$$

$$\text{temp} = 1 / 10 = 0$$

y

while ( $0 > 0$ )  $\Rightarrow$  false

if ( $0 * 1 = 0 = 100$ )  $\Rightarrow$  false.

exist for loop

for ( $i = 101$ ;  $i \geq 111$ ;  $i++$ )

continue all these process

to get  $i = 111$

prog 6: all divisors of a number

36  $\Rightarrow$  1, 2, 3, 4, 6, 9, 12, 18, 36

#include <iostream>

using namespace std;

int main ()

{

cout << "Enter number :- "

cin >> n;

cout << "All divisors of "

<< n << " are :- " << endl;

```
for (int i = 1; i <= n; i++)  
{  
    if (n % i == 0)  
    {  
        cout << i << endl;  
    }  
}  
  
return 0;  
}
```

→ output:-

Enter Number :- 36

All Divisors of :- 36 are :-

1  
2  
3  
4  
6  
9  
12  
13  
18

36

\* prog 7 :- Power Exponentiation.

$$x = 2$$

$$n = 5$$

$$\text{pow}(2, 5) = 2^5 = \underline{\underline{32}}$$

Now

$$2^{21} = 2 \cdot 2^{20}$$

$$2^{20} \Rightarrow (2^2)^{\frac{20}{2}} \Rightarrow (4)^{10}$$

$$\Rightarrow (4^2)^{\frac{10}{2}} \Rightarrow (16)^5$$

$$(16)^5 \Rightarrow 16 \times 16^4$$

$$16^4 \Rightarrow (16^2)^{\frac{4}{2}} \Rightarrow (256)^2$$

$$(256^2)^{\frac{2}{2}} \Rightarrow 65536$$

$$65536 = (65536) (65536)$$

$$\text{ans} = 1 \times 2 \times 16 \times 65536$$

$$\Rightarrow 2097152$$

$$= 2^{21}$$

```
#include <iostream>
using namespace std;
```

```
int powerFunction( int x, int n ) {
```

```
{
```

```
int ans = 1;
```

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

```
{
```

```
if ( n % 2 == 0 ) {
```

```
{
```

```
ans = ans * x;
```

```
n = n / 2;
```

```
}
```

```
else {
```

```
{
```

```
n = n / 2;
```

```
x = n * n;
```

```
1/n^4 = (x^2)^4/2
```

```
}
```

```
}
```

```
return ans;
```

```
}
```

```
int main () {
```

```
{
```

```
int x, n;
```

```
cout << "Enter Base Number :- ";
```

```
cin >> x;
```

```
cout << "Enter Exponent :- ";
```

```
cin >> n;
```

```
cout << powerFunction ( x, n );
```

```
return 0;
```

```
}
```

→ output :-

Enter Base Number :- 2

Enter Exponent :- 21

$2^{21} = 152$ .

→ Dry Run = 1 - 2 - 0 = 0

$$x = 2$$

$$n = 21$$

$$ans = 1$$

while ( $21 > 0$ )  $\Rightarrow$  true

{

    if ( $21 \% 2 != 0$ )  $\Rightarrow$  true.

{

$$ans = 1 * 2 = 2$$

$$n = 21 - 1 = 20$$

}

{

    while ( $20 > 0$ )

{

        if ( $20 \% 2 != 0$ )  $\Rightarrow$  false.

        else

{

$$n = 20 / 2 = 10 \text{ if } 0$$

$$n = 2 * 2 = 4$$

{

    while ( $10 > 0$ )

{

        if ( $10 \% 2 != 0$ )  $\Rightarrow$  false.

        else

$$n = 10 / 2 = 5$$

$$n = 5 * 4 = 16$$

{

vehicle ( $5 > 0$ )

{

if ( $5 \times 2 != 0$ ) = true.

$$\text{ans} = 2 * 16 = 32$$

$$n = 5 - 1 = 4$$

}

vehicle ( $4 > 0$ )

{

if ( $4 \times 2 != 0$ ) = false.

else

~~$n = 4 / 2 = 2$~~

~~$x = 2 * 2 = 16 * 16$~~

~~$= 256$~~

}

{

vehicle ( $2 > 0$ )

{

if ( $2 \times 2 \neq 0$ ) = false

else

~~$n = 2 / 2 = 1$~~

~~$x = 256 * 256$~~

~~$= 65536$~~

}

{

vehicle ( $1 > 0$ )

{

if ( $1 \times 2 \neq 0$ ) = true

~~$\text{ans} = 32 * 65536$~~

~~$= 2097152$~~

$$n = 1 - 1 = 0$$

Date :

Page No.

while ( $0 > 0$ )  $\Rightarrow$  false.

print (2097152).