

লুপ

লুপ হচ্ছে একটি কাজের পুনরাবৃত্তি।

লুপ ও পদ্ধতি -

1. for loop

2. while loop

3. do while loop

→ Almost same



একটি লুপের মুক্তি হ্যাভি

1. Start (শুরু)

2. Condition / Relational Operators (অর্থ)

3. Statement (ক্রিয়তা / কাজ)

4. increment / decrement (যোগ / ঘোম)

লুপের কার্যপদ্ধতি:

1. কর্ত যাচাই করবে

2. কর্ত সত্ত্ব হলে কাজ করবে

3. দুঃখি বা হাম করব

4. $i = i + 1$

old value → new value

= assignment operators

Subject.....
Date:..... Time:.....

▣ $i=0$ এবং মানু থলো । এবং মান 0 হিসেবে

assign ক্ষেত্ৰটোকুন্দ চৰাকুল পুঁজি বৰুৱা গুৰি

ব) $i = i+1$ এর মানে হলো— i এর পূর্বের মানের
মাঝে 1-যোগ ক্ষণ।

$i = i + 2$ এর মানে হলো i এর পূর্বে ~~পুরো~~ মানের-
মাঝে ২ যোগ করা।

$i = i + 3$ এর মানে- হলো i এর পূর্বে- মানের মাঝে
৩ স্থাগ করা।

$i = i + 4$ ~~93~~ " ikarrikolad " mifibromi.

৪ মেগ ক্ষয়াট (আর) (মেচি) *freelancer*.

(1878) 1878) *franciscus* \ *francisi*.

Postfix :

int i = 65;

~~printf("y.%d", i++);~~ 65 ~~+ + i~~ prefix (i = i+1)

printf ("%d", i); 66

++ postfix

1000 → 2000

Prefix:

inf $i=65$ the wrong

```
printf ("%d", ++i); 66
```

printf ("%d", i); 66

While Loop এবং পর্যন্ত:

for loop ଏବଂ ମୂଳ୍ୟ:

while \rightarrow

10

11

ପାତ୍ର
କାଳି
ମହିଳା

while (~~ans~~)

বিষ্ণু / বিষ্ণু

খণ্ডি / খণ্ডি

(red) b) 2nd

$\lim_{n \rightarrow \infty} b_n = 0$

(Signature)

30 minutes

Mimhaaz Sir's Note

Subject.....
Date:..... Time:.....

Increment

$i++$ postfix
 $++i$ prefix $\Rightarrow \underline{(i = i + 1)}$

অঙ্গিশ্চরণ	সর্ব রূপ
$i++$	$i = i + 1$
$++i$	$i = i + 1$
$i + 1$	$i = i + 1$
$i + 2$	$i = i + 2$
$i + 3$	$i = i + 3$
$i + 4$	$i = i + 4$
$i + 5$	$i = i + 5$
$i + 6$	$i = i + 6$
$i + 7$	$i = i + 7$
$i + 8$	$i = i + 8$

Note:

$i = i + 1$
 $\downarrow \rightarrow$ প্রক্রিয়া
 সমান মান
 এবং মান, i এবং মান
 ১ ক্রমে পুনরি ক্রয়।
 $i = i + 2$; i এবং মান
 ২ ক্রমে পুনরি ক্রয়।
 $i = i + 3$ এবং মান,
 i এবং মান ৩ ক্রমে পুনরি ক্রয়।
 এই ক্ষেত্রে, কোন ক্ষিতি
 হাম ইলে + এবং জয়গাম
 - হবে।

*

1 2 3 4

#include <stdio.h>
int main() {
 int i;
 i = 1;
 while (i <= 100) {
 printf("%d", i);
 i++;
 }
 return 0;
}

*

1 2 3 4

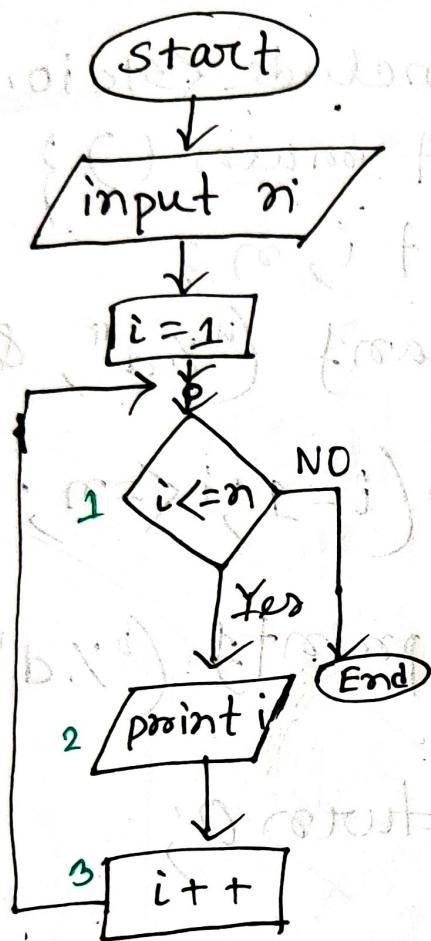
#include <stdio.h>
int main() {
 int i, n;
 scanf("%d", &n);
 i = 1;
 while (i <= n) {
 printf("%d", i);
 i++;
 }
 return 0;
}

#include <stdio.h>
int main() {
 int i;
 for (i = 1; i <= 100; i++)
 printf("%d", i);
 return 0;
}

#include <stdio.h>

int main() {
 int i, n;
 scanf("%d", &n);
 for (i = 1; i <= n; i++)
 printf("%d", i);
 return 0;
}

Subject.....
Date:..... Time.....



Step 01 : Start

Step 02 : input n

Step 03 : $i = 1$

Step 04 : is $i \leq n$

i. Yes, go to step 05

ii. No, go to step 07

Step 05 : print i

Step 06 : $i++$, go to

Step 04

Step 07 : End

Subject.....

Date:..... Time.....

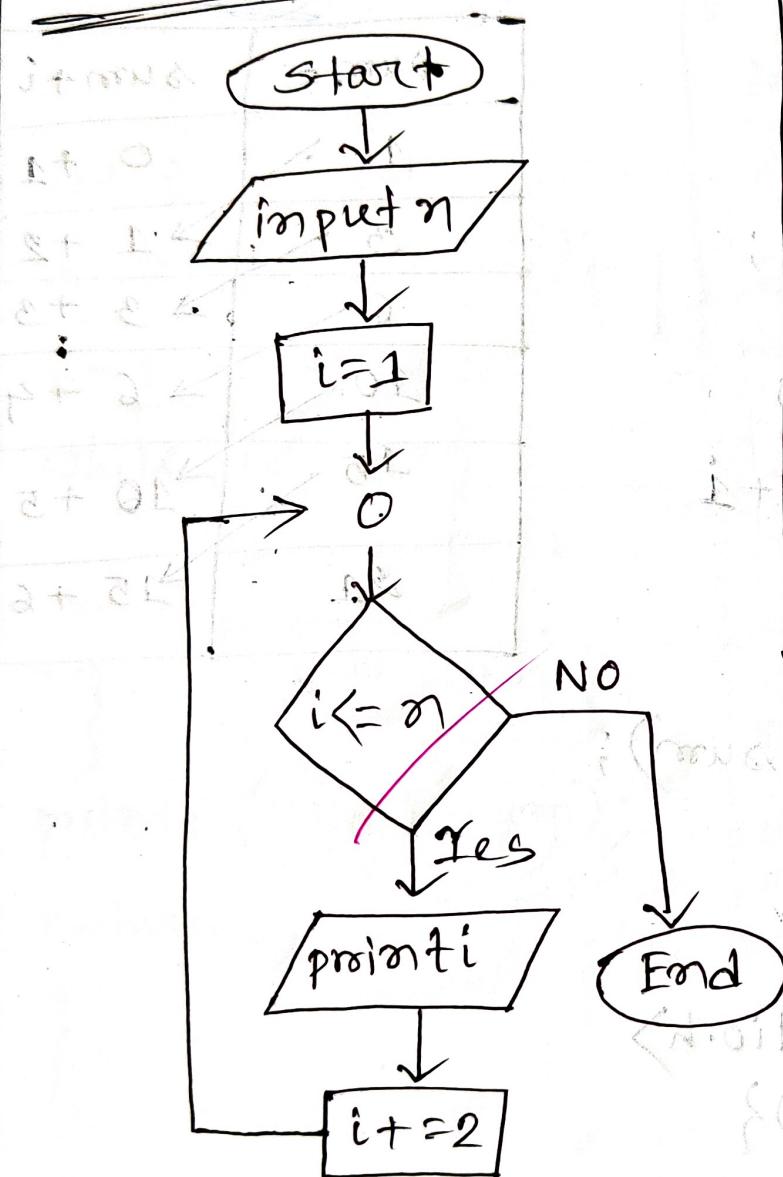
① 1 to 5 Laps

```
#include <stdio.h>
int main()
{
    int i; // input
    scanf ("%d", &i);
    do gate i=1 to n:
        if while (i <= n)
            printf ("%d", i);
        i+=2; // 0 gate
    } // gate
    return 0;
}
```

```
#include <stdio.h>
int main()
{
    int i; // input
    scanf ("%d", &i);
    for (i=1; i<=n; i+=2)
        printf ("%d", i);
    return 0;
}
```

Subject.....
Date:..... Time:.....

Flowchart



Algorithm

Step 01: start
Step 02: input n
Step 03: $i = 1$
Step 04: $i \leq n$
i. Yes, go to step 05
ii. NO, go to step 07
Step 05: print i
Step 06: $i = i + 2$, go to step 04
Step 07: End

$$\text{Q1} 1 + 2 + 3 + \dots + n$$

(Ans)

```
#include <stdio.h>
int main() {
    int i, n, sum;
    scanf("%d", &n);
    i = 1, sum = 0;
    while (i <= n) {
        sum = sum + i
        i++;
    }
    printf("%d", sum);
    return 0;
}
```

$\Delta \text{sum} =$	$\text{sum} + i$
1	0 + 1
3	1 + 2
6	3 + 3
10	6 + 4
15	10 + 5
21	15 + 6

```
#include <stdio.h>
int main() {
    int i, n, sum;
    scanf("%d", &n);
    sum = 0;
    for (i = 1; i <= n; i++)
        sum = sum + i;
    printf("%d", sum);
    return 0;
}
```

Subject.....
Date..... Time.....

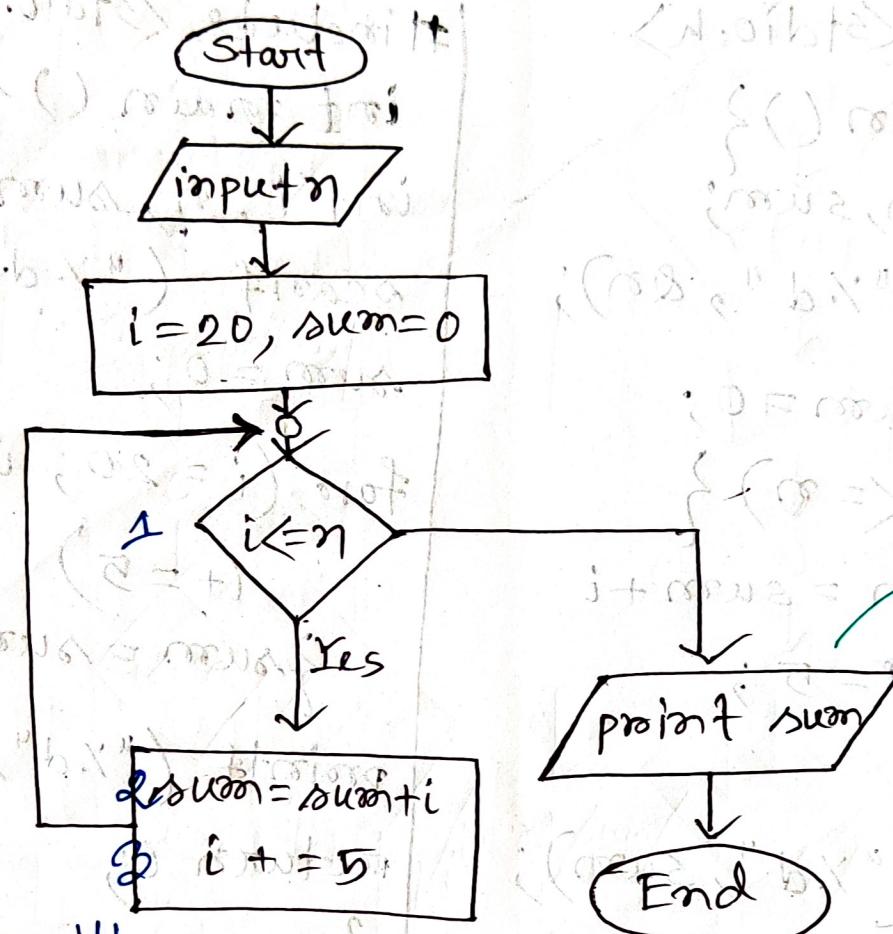
$$20 + 25 + 30 + \dots + n$$

```
#include <stdio.h>
int main () {
    int i, n, sum;
    scanf ("%d", &n);
    i = 20, sum = 0;
    while (i <= n) {
        sum = sum + i;
        i += 5;
    }
    printf ("%d", sum);
    return 0;
}
```

```
#include <stdio.h>
int main () {
    int i, n, sum;
    scanf ("%d", &n);
    sum = 0;
    for (i = 20; i <= n; i += 5)
        sum = sum + i;
    printf ("%d", sum);
    return 0;
}
```

frate : 10 gate
is busni : 20 gate
0 = max, 0.2 = 1.20 gate
 \Rightarrow 1.20 gate
20 gate at op. val. 0

Flowchart



Algorithm

- Step 01 : start
 Step 02 : input n
 Step 03 : $i = 20, sum = 0$
 Step 04 : is $i \leq n$
 - i. Yes, go to step 05
 - ii. No, go to step 07
 Step 05 : $sum = sum + i$
 Step 06 : $i = i + 5$, go to step 04
 Step 07 : print sum
 Step 08 : End

* $3 + 6 + 9 + \dots + n$

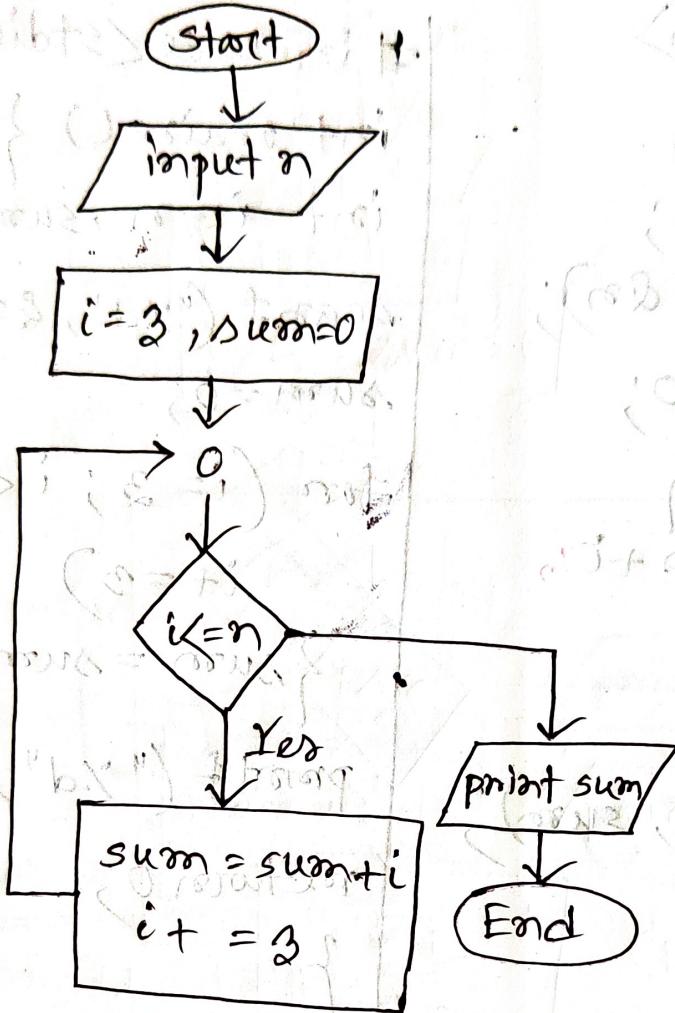
#include <stdio.h>

```
int main () {  
    int i, n, sum;  
    scanf ("%d", &n);  
    i = 3, sum = 0;  
    while (i <= n) {  
        sum = sum + i;  
        i += 3;  
    }  
    printf ("%d", sum);  
    return 0;  
}
```

#include <stdio.h>

```
int main () {  
    int i, n, sum;  
    scanf ("%d", &n);  
    sum = 0;  
    for (i = 3; i <= n; i += 3)  
        sum = sum + i;  
    printf ("%d", sum);  
    return 0;  
}
```

Flowchart



Algorithm

- Step 01 : Start
- Step 02 : input n
- Step 03 : $i=3, sum=0$
- Step 04 : is $i \leq n$
 - i. Yes, go to Step 05
 - ii. No, go to step 07
- Step 05 : $sum = sum + i$
- Step 06 : $i = i + 3$, go to step 4
- Step 07 : print sum
- Step 08 : End

$$1 + 2^r + 3^r + \dots + n^r$$

or

```
#include <stdio.h>
int main() {
    int i, n, sum;
    scanf ("%d", &n);
    i = 1, sum = 0;
    while (i <= n) {
        sum = sum + i * i;
        i++;
    }
    printf ("%d", sum);
    return 0;
}
```

return 0;

}

For loop:

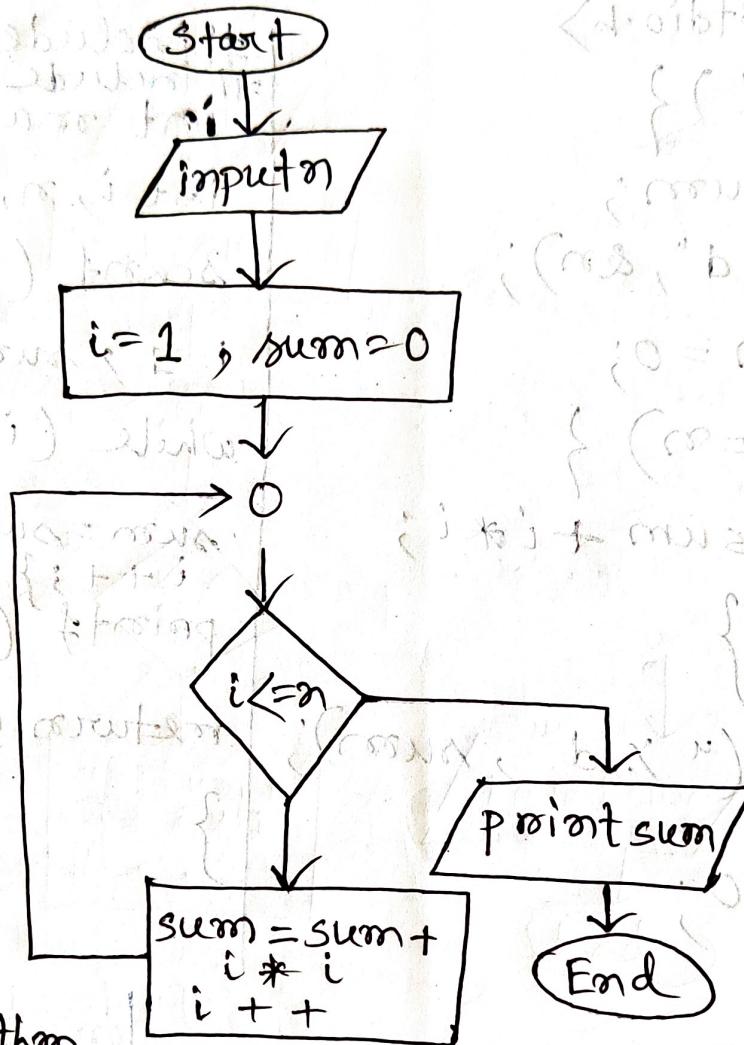
```
#include <stdio.h>
int main() {
    int i, n, sum;
    scanf ("%d", &n);
    sum = 0;
    for (i = 1; i <= n; i++)
        sum = sum + i * i;
    printf ("%d", sum);
    return 0;
}
```

```
#include <stdio.h>
#include <math.h>
int main() {
    int i, n, sum;
    scanf ("%d", &n);
    i = 1, sum = 0;
    while (i <= n) {
        sum = sum + pow(i, 2);
        i++;
    }
    printf ("%d", sum);
    return 0;
}
```

or

```
#include <stdio.h>
#include <math.h>
int main() {
    int i, n, sum;
    scanf ("%d", &n);
    sum = 0;
    for (i = 1; i <= n; i++)
        sum = sum + pow(i, 2);
    printf ("%d", sum);
    return 0;
}
```

Flow Chart



Algorithm

- Step 01: Start
- Step 02: input
- Step 03: $i=1, sum=0$
- Step 04: is $i \leq n$
 - i. Yes, go to Step 05
 - ii. No, go to step 07
- Step 05: $sum = sum + i * i$
- Step 06: $i++,$ go to step 04
- Step 07: print sum
- Step 08: End

$$\textcircled{*} 1^3 + 2^3 + 3^3 + \dots + n^3$$

```
#include <stdio.h>
int main () {
    int i, n, sum;
    scanf ("%d", &n);
    i = 1, sum = 0;
    while (i <= n) {
        sum = sum + i * i * i;
        i++;
    }
    printf ("%d", sum);
    return 0;
}
```

↓
break

For Loop:

```
#include <stdio.h>
int main () {
    int i, n, sum;
    scanf ("%d", &n);
    sum = 0;
    for (i = 1; i <= n; i++)
        sum = sum + i * i * i;
    printf ("%d", sum);
    return 0;
}
```

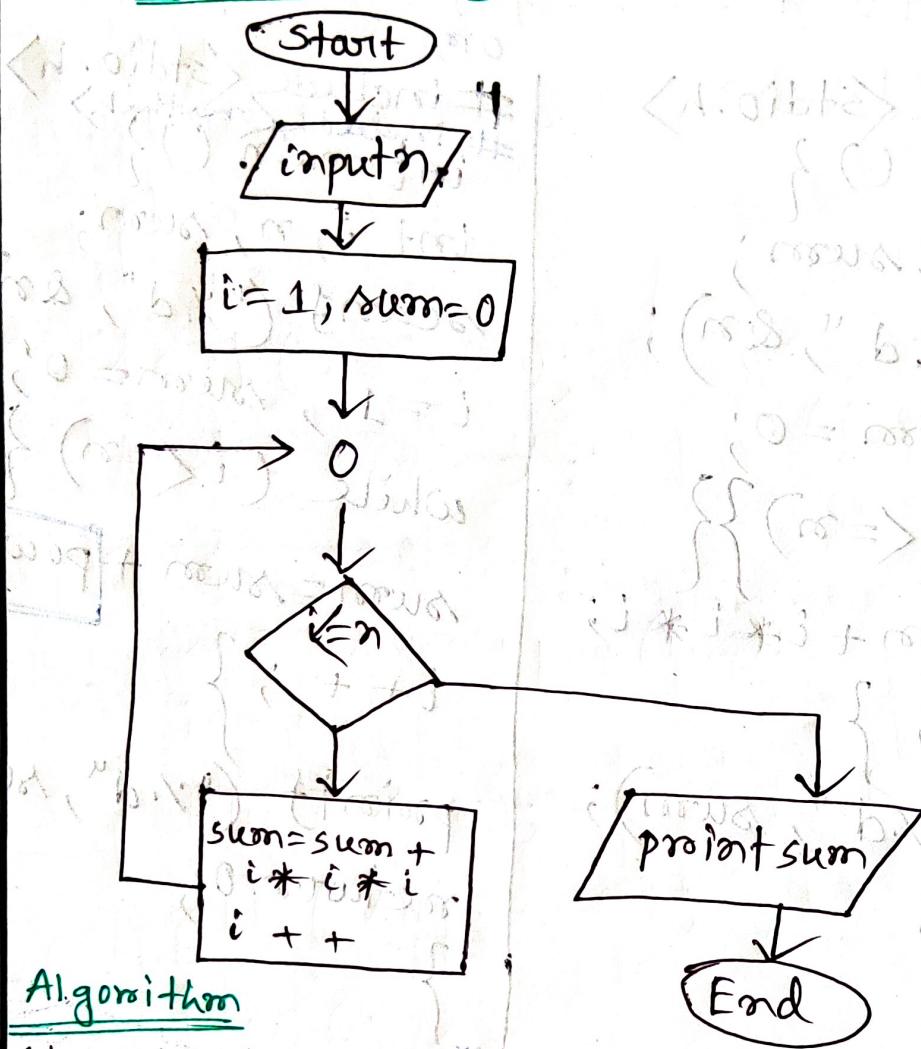
```
#include <stdio.h>
#include <math.h>
int main () {
    int i, n, sum;
    scanf ("%d", &n);
    i = 1, sum = 0;
    while (i <= n) {
        sum = sum + pow(i, 3);
        i++;
    }
    printf ("%d", sum);
    return 0;
}
```

↑ Pow
get
math

```
#include <stdio.h>
#include <math.h>
int main () {
    int i, n, sum;
    scanf ("%d", &n);
    sum = 0;
    for (i = 1; i <= n; i++)
        sum = sum + pow(i, 3);
    printf ("%d", sum);
    return 0;
}
```

↑ Pow
get
math

Flow chart



Algorithm

Step 01 : start
Step 02 : input number
Step 03 : $i = 1, sum = 0$
Step 04 : is $i \leq n$
 i. Yes, go to step 05
 ii. No, go to step 07

Step 05 : $sum = sum + i * i$
Step 06 : $i++,$ go to step 04

Step 07 : print sum

Step 08 : End

~~Postfix :~~

```
int i, k;
```

```
i = 7;
```

```
k = i++;
```

```
printf ("%d %d", i, k);
```

```
K=7, i=8
```

i = i + 1

Prefix :

```
int i, k;
```

```
i = 7;
```

```
k = ++i;
```

```
printf ("%d %d", i, k);
```

```
K=8, i=8
```

i = i + 1

do while loop এর

পর্যায় :

মুক্তি ;

do {

বিবৃতি / বশজি ;

বৃদ্ধি / অগ্রসর ;

} while (অর্থ) ;

do while এ কাজ করার
পথ গেরে

do while

Subject.....
Date..... Time.....

✳ $2^r + 4^r + 6^r + \dots + n^r$

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

```
#include <math.h>
```

```
int main () {
```

```
int i, n, sum;
```

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

```
i = 2, sum = 0;
```

```
do {
```

```
sum = sum + pow (i, 2);
```

```
i += 2;
```

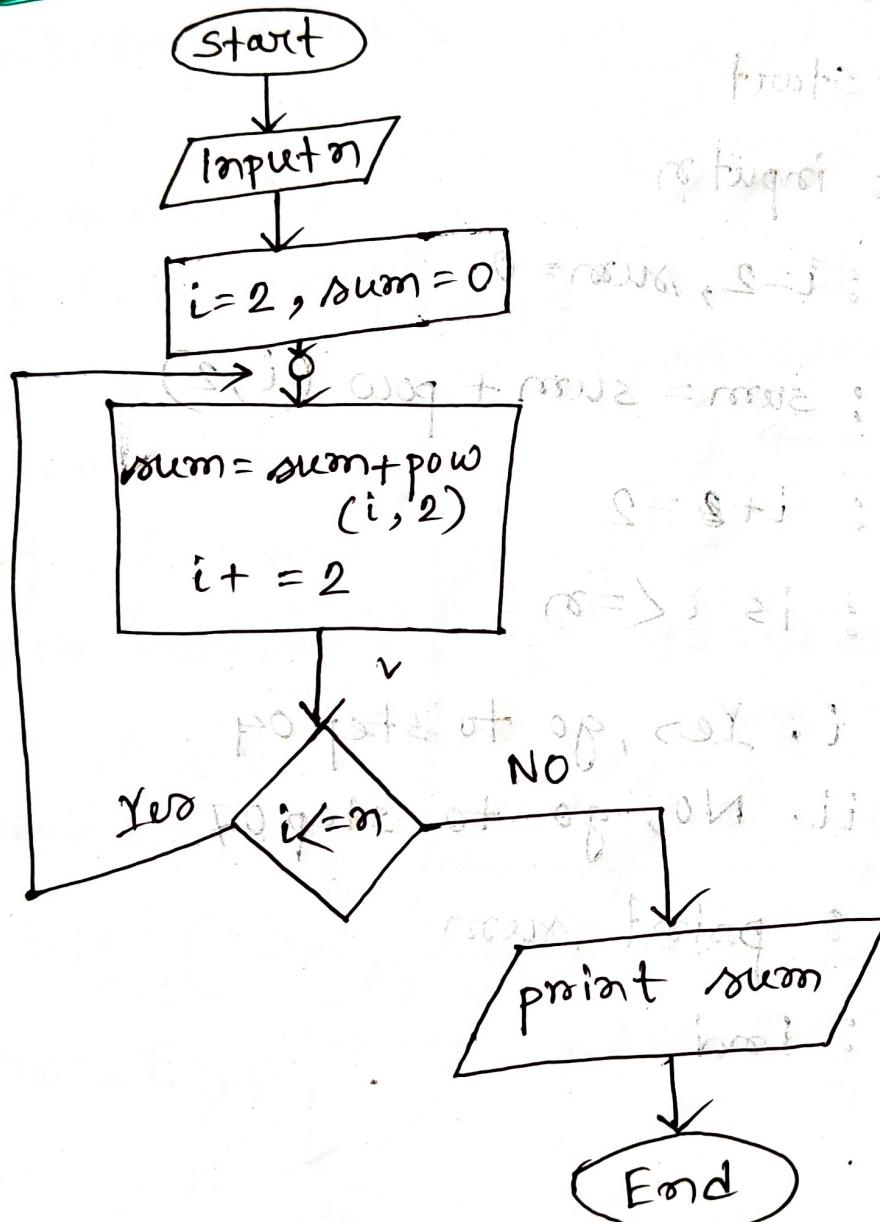
```
} while (i <= n);
```

```
printf ("%d", sum);
```

```
return 0;
```

```
}
```

Flowchart



Note: পরীক্ষার যাত্রা C-programme লিখতে

বললে— for loop বা while loop লিখত হবে।

Algorithm:

Step 01 : Start

Step 02 : input n

Step 03 : $i=2, sum = 0$

Step 04 : $sum = sum + pow(i, 2)$

Step 05 : $it = 2$

Step 06 : is $i \leq n$

i. Yes, go to step 04

ii. No, go to step 07

Step 07 : print sum

Step 08 : End

$$10 + 16 + 22 + \dots + 70$$

#include <stdio.h>

```
int main() {
```

```
int i, n, sum;
```

```
i = 10, sum = 0;
```

```
do {
```

```
sum = sum + i;
```

```
i = i + 6;
```

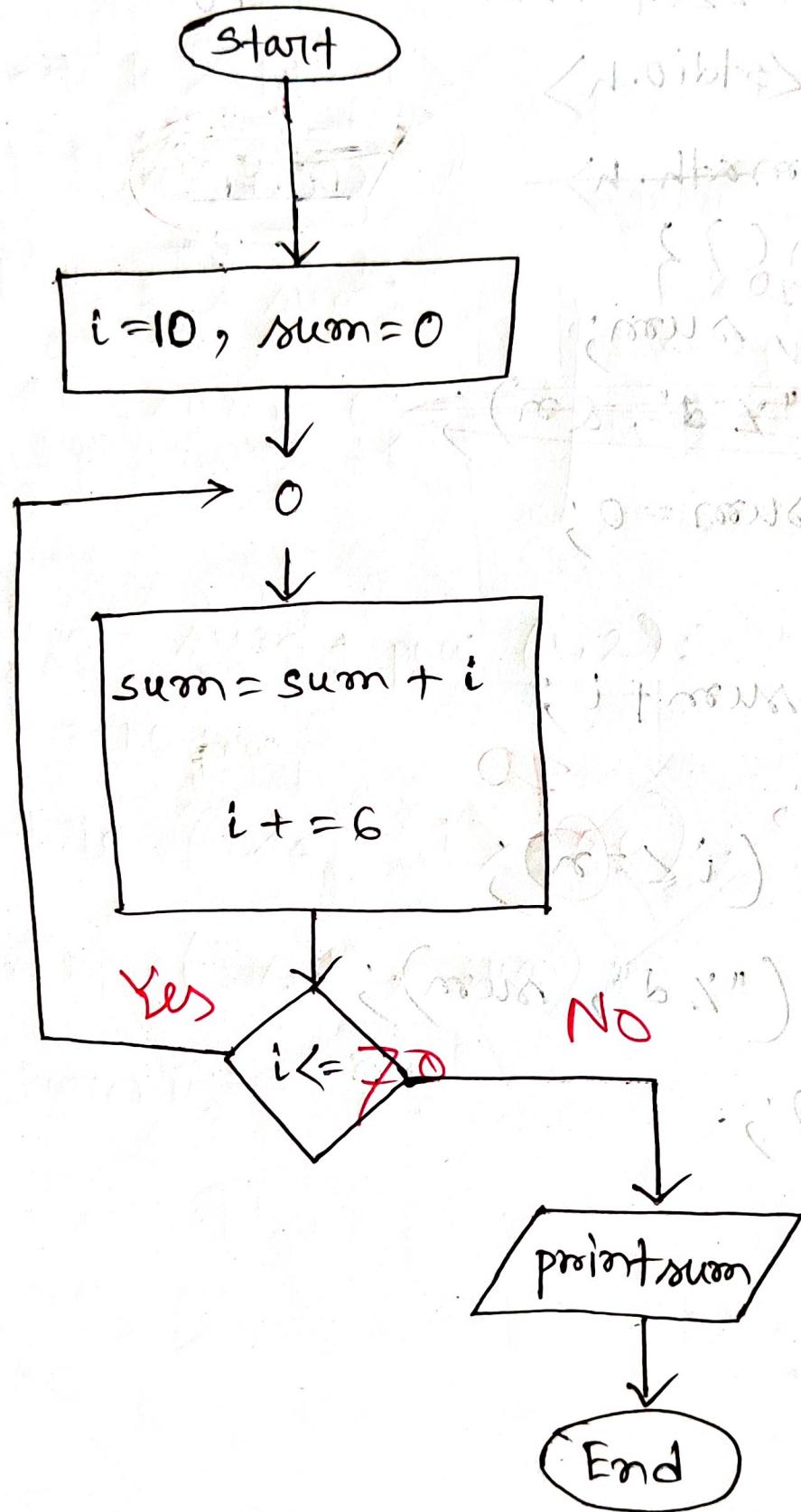
```
} while (i <= n);
```

```
printf ("%d", sum);
```

```
return 0;
```

```
}
```





Step 01 : Start

Step 03 : $i = 10$, $sum = 0$ f() returns 1

Step 04 : $sum = sum + i$ (now 00 \rightarrow 10)

Step 05 : $i + 1 = 6$ $10 = 001010_2$

Step 06 : is $i <= 70$ $\cancel{10}$

i. Yes, go to step 04

ii. No, go to step 07

Step 07: print sum

Step 08: End

(*) $90^{\circ} + 80^{\circ} + 70^{\circ} + \dots + 20^{\circ}$ का मान

#include <stdio.h>

int main () {

int i, sum;

i = 90, sum = 0;

do {

sum = sum + pow (i, 2);

i -= 10;

} while (i >= 20);

printf ("%d", sum);

return 0;

}

Step 01: Start
Step 02: $i = 90$, $sum = 0$

Step 03: $sum = sum + pow(i, 2)$

Step 04: $i = i - 10$

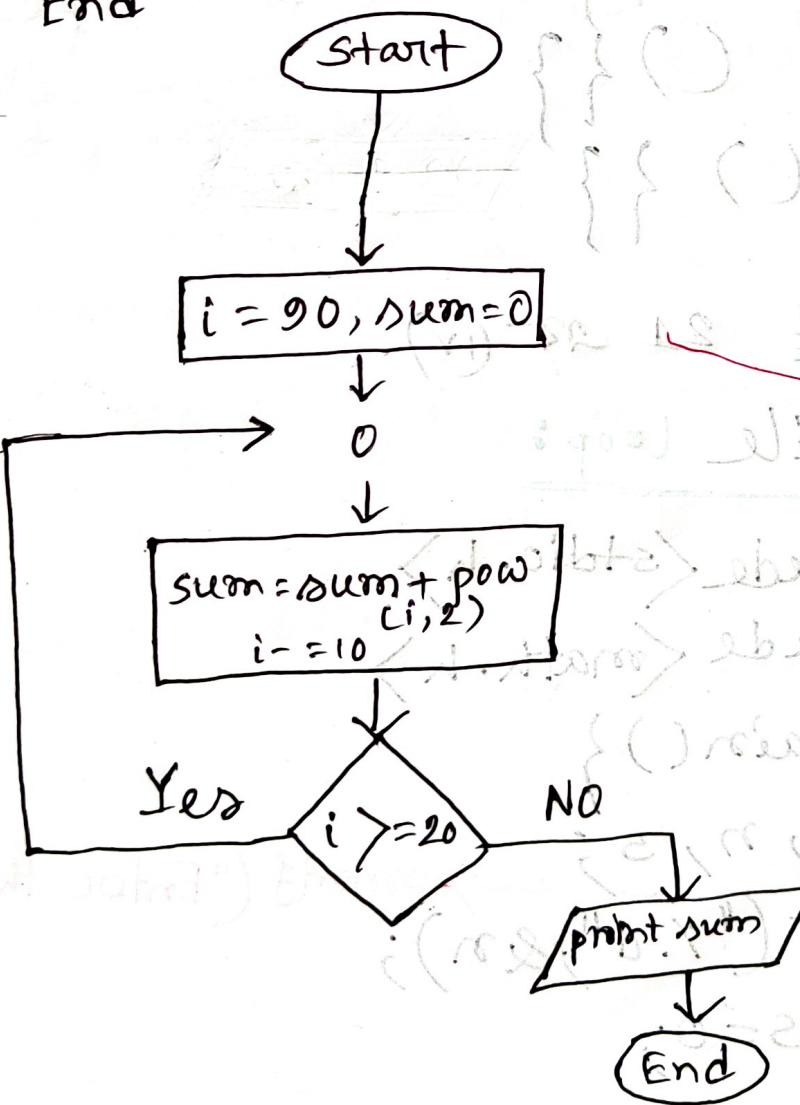
Step 05: If $i >= 20$

i. Yes, go to Step 03

ii. No, go to Step 06

Step 06: print sum

Step 07: End



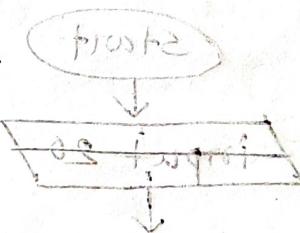
সুমিত্রেলন দিতে হবে

```
printf();  
scanf();  
do {} while();
```

for loop এর মধ্যে
যা সাধা গহণ করে

সুমিত্রেলন হবে না

```
main () {}  
for () {}  
while () {}  
if () {}
```



HW: 21 জু (iv)

do while loop:

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

```
int main()
```

```
int i, n, s;
```

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

```
i=5, s=0;
```

```
do {
```

```

s = s + pow(i, 3);
i++;
} while (i <= n);
printf("%d", s);
return 0;
}

while loop :
#include <stdio.h>
#include <math.h>
int main() {
    int i, n, s;
    scanf("%d", &n);
    i = 5, s = 0;
    while (i <= n) {
        s = s + pow(i, 3);
        i++;
    }
    printf("%d", s);
    return 0;
}

```

ଲୁଗାଳ୍ ନିଯମନ

ଲୁଗାଳ୍ ୨ ଏବେ ନିଯମନ କର୍ତ୍ତା ମାତ୍ର -

(i) continue → ଏକିଥି ମାତ୍ରୋ

(ii) break → ଆମା

continue:

```
for(i=1; i<=7; i++) {
```

```
    if (i==4) continue;
```

```
    printf ("%d", i);
```

```
}
```

output:

1 2 3 5 6 7

break:

```
for(i=1; i<=7; i++) {
```

```
    if (i==4) break;
```

```
    printf ("%d", i);
```

```
}
```

output:

1 2 3

break:

```
for (i=1; i<=7; i++) {
    printf ("%d", i);
    if (i==4) break;
}
```

Output:

1 2 3 4

Note: break এর মত প্রয়োগ করেই - লুপ থেকে
যাবে, break printf এর আগে থাকলে - আপে
break থাবাও এবং printf এর পর থাকলে - পারে
break থাবা।

28/5/

```
#include <stdio.h>
#include <math.h>
#include <conio.h>
void main ()
{
    int i, n, sum;
    printf ("Enter the value of n:");
    scanf ("%d", &n);
    i = 1, sum = 0;
    do {
        if (i == 3) continue;
        sum = sum + pow (i, 3);
        i = i + 1;
    } while (i <= n);
    printf ("%d", sum);
    getch ();
}
```

$$(*) 1^2 + 2^2 + 4^2 + 5^2 + \dots + n^2$$

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

```
int main () {
```

```
    int i, n, sum;
```

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

```
    sum = 0;
```

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

```
        if (i == 3) continue;
```

```
        sum = sum + i * i;
```

```
        printf ("%d", i);
```

```
    return 0;
```

```
}
```

Subject.....

Date..... Time.....

go to লপ প্রয়োগ করা।

গুরুত্ব:

ক্ষমতা;

নেইল:

বিবৃতি / শোভা;

বৃদ্ধি / দ্রাঘি;

if (শর্ত) goto নেইল; } while (শর্ত);

⊗ 5^r + 8^r + 11^r + + n^r

#include <stdio.h>

int main() {

int i, s, n;

scanf ("%d", &n);

i = 5, s = 0;

A:

s = s + i * i;

i += 3;

if (i <= n) goto A;

printf ("%d", s);

return 0;

}

⊗ $2^r + 4^r + 6^r + \dots + 80^r$

#include <stdio.h>

int main () {

int i, sum;

i = 2, sum = 0;

SABA:

sum = sum + i * i;

i = 2;

if (i <= 80) go to SABA;

printf ("%d", sum);

return 0;

}

Subject.....
Date:..... Time.....

Test (2024)

7. (g)

Step 01: Start

Step 02: input n

Step 03: $K = 6, \text{sum} = 0$

Step 04: is $K \leq n$

i. Yes, go to step 05 \Rightarrow (2+8) result

ii. No, go to step 07

Step 05: $\text{sum} = \text{sum} + K$

Step 06: $K = K + 6$, go to step 04

Step 07: print sum

Step 08: End

Q) #include <stdio.h>

int main()

{

int K, sum;
scanf("%d", &n);
sum=0;

for(K=6, K<=n; K=K+6)

{ sum = sum + K; }

printf("The sum of the series is

"%.d", sum);

return 0;

}

$1 \times 2 \times 3 \times 4 \times \dots \times n$

while Loop ↗

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

```
int main () {
```

```
int i, n, s;
```

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

```
i=1, s=1;
```

```
while (i<=n) {
```

```
s=s * i;
```

```
i++;
```

```
}
```

```
printf ("%d",
```

```
s);
```

```
return 0;
```

```
}
```

For Loop ↗

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

```
int main () {
```

```
int i, n, sum;
```

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

```
sum=1;
```

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

```
{sum=sum * i;}
```

```
printf ("%d", sum);
```

```
return 0;
```

```
}
```

ফাক্টোরিয়েল নির্ণয়:

$$7! = 1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7$$

$$5! = 1 \times 2 \times 3 \times 4 \times 5$$

ক্লেন অংশীয় - ফাক্টোরিয়েল হচ্ছে 1 থেকে এ অংশীয়

পর্যন্ত ক্রমিক অংশীয় পুনরাবৃত্তি।

(*) $2^3 + 3^4 + 4^5 + 5^6 + 6^7 + \dots$ in terms of n

```
#include <stdio.h>
#include <math.h>
int main () {
    int i, n, s;
    scanf ("%d", &n);
    i = 2, s = 0;
    while (i <= n) {
        s = s + pow(i, i+1);
        i++;
    }
    printf ("%d", s);
    return 0;
}
```

```
#include <stdio.h>
#include <math.h>
int main () {
    int i, n, s;
    scanf ("%d", &n);
    s = 0;
    for (i = 2; i <= n; i++)
        s = s + pow(i, i+1);
    printf ("%d", s);
    return 0;
}
```

গ.স.গ. (G.C.D.)

$$\begin{array}{r}
 24) 40(1 \\
 \underline{24} \\
 16) 24(1 \\
 \underline{16} \\
 8) 16(2 \\
 \underline{16} \\
 0
 \end{array}$$

মুক্ত সংখ্যার হেট সংখ্যা দিয়ে গু করো
যাচ্ছেন না — শুণ্য হয়,

$$\begin{array}{r}
 40) 64(1 \\
 \underline{40} \\
 24) 40(1 \\
 \underline{24} \\
 16) 24(1 \\
 \underline{16} \\
 8) 16(2 \\
 \underline{16} \\
 0
 \end{array}$$

While loop କ୍ଷେତ୍ର କ୍ଷେତ୍ର G.C.D. ନିର୍ଣ୍ଣୟ:

#include <stdio.h>

```
int main () {
```

```
    int small, large, r;
```

```
    scanf ("%d %d", &small, &large);
```

```
    while (large % small != 0) {
```

```
        r = large % small;
```

```
        large = small;
```

```
        small = r;
```

```
}
```

```
    printf ("%d", small);
```

```
    return 0;
```

```
}
```

r
↓
remainder
↓
GCD

40, 24	24, 16	16, 8
$r=16$	$r=8$	$r=0$
large = large		
$24 = 16 + 8$		
small = small		
$16 = 8 \times 2$		
		$= 8$

64, 40	40, 24	24, 16	16, 8
$r=24$	$r=16$	$r=8$	$r=0$
large = 40	large = 24	large = 16	
small = 24	small = 16	small = 8	

$$17) \frac{11}{7} (1)$$

$$\pi(8) 4) \frac{1}{7} (1)$$

$$\pi(8) \frac{3}{3) 4 (1}$$

{ (plus $\frac{3}{3}$) balance & repeat } (do while)

$$\pi(8) 1) \frac{3}{3) 3$$

$$\underline{0}$$

(do while)

#include <stdio.h>

int main () {

int s, r;
scanf("%d", &r);
do {

r = r % s;

l = l + 1;

s = r;

} while (l % s != 0);

printf ("%d", s);

return 0;

}

ଆମେ: ଏହାର ସମ୍ବନ୍ଧରେ କୁଟୀ ଚାର୍ଟିଲେଖ ପ୍ରକଳ୍ପରେ ଆମେ ବଳା

२४।

int a;
int a0, a1, a2, ..., a1394;
= int a [1395];
↓
Data
Type Array Name Array Size
= int a[0], a[1], a[2], ..., a[1394];

Note: int m[10] এর মত, m[10]/m[11] চলে

ମନ୍ତ୍ରବ ନମ୍ବ

float b [25];

* আরে ডাইমেনশন:

ক. ১ ডাইমেনশন বা লিমিয়ার আরে:

উদা: int a[8];

খ. মালি- ডাইমেনশন বা লিমিয়ার আরে:

লিমিয়ার ২ প্রশ্ন:

i. 2D Array (int b [10] [15];)

ii. 3D Array (int b [10] [25] [7];)

এই নিচের প্রশ্ন আরেটি কী?

i. int a [3] [4];

ii. int p [3] {5};

ফাংশন:

প্রয়োগু- এখন- কতগুলো statement হুণ- নামে
একটি কৃতি গুরুত্ব অর্থে বেছান- শব- হুণ- নির্দিষ্ট
কোড সম্পাদন কর গুরুত্ব ফাংশন বল।

```
int main() {  
}
```

```
    a b c d . () {} }  
    m () {} }  
    Saba () {} } → এই হচ্ছে Saba
```

Programming এ নতুন শব্দ :

#include <stdio.h>

```
int main () {  
    printf ("Ajmena \n Ajmaan");  
    return 0;  
}
```

Ajmena
Ajmaan

HTML এ \n টিপ করে নতুন লাইন পিচ করা হবে।

Programming এ '\t' নতুন লাইন পিচ করে,

> টাব ফিল্ড আন - :

এটা টাব ফিল্ড সব ফিল্ড জ্যান টেবিল করা।

#include <stdio.h>

```
int main () {  
    printf ("Safwan \t Ahmed");  
    return 0;  
}
```

Safwan Ahmed

Subject.....

Date:..... Time.....



$$120, 80 = 40 \text{ (gcd)}, 240 \text{ (lcm)}$$

$$64, 40 = 8 \text{ (gcd)}, 320 \text{ (lcm)}$$

$$\boxed{120 \times 80 = 9600}$$
$$40 \times 240 = 9600$$

$$\boxed{64 \times 40 = 2560}$$
$$8 \times 320 = 2560$$

Note:

ଦ୍ୱାରା ହଟି ଯୁଧ୍ୟାର ପୁନଫଳ— ଲମ୍ବର ଗ.ନ୍ତ୍ରାସ୍

ଏବଂ ଲମ୍ବାସ୍ ଏବଂ ପୁନଫଳର ଯମାନ,

$$\boxed{\text{gcd} * \text{lcm} = a * b}$$

$$\text{lcm} = (a * b) / \text{gcd};$$

ଅନେକବୀରୀ
ଦ୍ୱାରା ହଟି ଯୁଧ୍ୟା

a ଓ b

ବ୍ୟାକ୍ସନ - ୨ ଶାଖାର ଲମ୍ବାତ୍ମନ ନିଯମ:

```
#include <stdio.h>

int main () {
    int small, large, r, a, b, lcm;
    scanf ("%d %d", &small, &large);
    a = large, b = small;
    while (large > small != 0) {
        r = large % small;
        large = small;
        small = r;
    }
    lcm = (a * b) / small;
    printf ("%d", lcm);
    return 0;
}
```