

C++

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

struct Node {
    int data;
    Node *next1;
    Node *next2;
    Node(int x)
        { data = x; }
};

void print1(Node *head)
{
    Node *curr = head;
    while (curr)
        cout << curr->data << " ";
    curr = curr->next1;
}

void print2(Node *head)
{
    Node *curr = head;
    while (curr)
        cout << curr->data << " ";
    if (curr->next2 != NULL)
        curr = curr->next2;
    else
        curr = curr->next1;
}
```

// Creating linked list 1

```
Node * makeL (int n, int limit)
{
    Node * head = new Node (^);
    Node * curr = head; int x = n + ^;
    while (x <= limit)
    {
        curr->next = new Node (x);
        x = x + 1;
        curr = curr->next;
    }
    return head;
}
```

Creating linked list 2

```
Node * makeC (int b, int n, int c, Node * l)
{
    Node * head2 = NULL;
    Node * curr2 = head2;
    int x = b;
    int y1 = c;
    Node * curr1 = head1;
    if (b == c)
    {
        while (curr1->data != c)
        {
            curr1 = curr1->next;
        }
    }
}
```

flag = 0

$b += x;$

$c += y;$

$\text{head}^2 = \text{curr}^1;$

$\text{curr}^2 = \text{head}^2;$

}

else

{ $\text{head}^2 = \text{newNode}(b);$

$\text{curr}^2 = \text{head}^2;$

$b += x;$

}

int flag = 0;

while ($b < n$)

{ if ($b == c$)

{ while ($\text{curr}^1 \rightarrow \text{data} != c$)

{ $\text{curr}^1 = \text{curr}^1 \rightarrow \text{next}^1;$

}

$\text{curr}^2 \rightarrow \text{next}^1 = \text{curr}^1;$

$\text{curr}^2 = \text{curr}^2 \rightarrow \text{next}^1;$

$b += n;$

$c += y;$

$\text{flag} = 1;$

}

else

{ if ($\text{flag} == 1$)

{ $\text{curr}^2 \rightarrow \text{next}^2 = \text{newNode}(b);$

$\text{curr}^2 = \text{curr}^2 \rightarrow \text{next}^2;$

$\text{flag} = 0;$

C++

b+ = b;

{
else{
curr2->next1 = new node(b);
b+ = b;

{

{

{
returns head2;

{

// printing them in sorted fashion

void merge(node *head1, node *head2)

{
node *curr1 = head1;

node *curr2 = head2;

while(curr1 != NULL && curr2 != NULL)

{
if(curr1->data < curr2->data){
cout << curr1->data << " ")

curr1 = curr1->next1;

{

elseif(curr2->data < curr1->data)

{
cout << curr2->data << " ")

curr2 = curr2->next1;

{

```
else if (curr1 == curr2)
```

```
{ cout << curr1->data << " ";
    curr1 = curr1->next;
    curr2 = curr2->next2;
```

{

```
}
```

```
while (curr1 != NULL)
```

```
{ cout << curr1->data << " ";
    curr1 = curr1->next;
```

{

```
while (curr2 != NULL)
```

```
{ cout << curr2->data << " ";
    if (curr2->next2 == NULL)
```

```
    curr2 = curr2->next2;
    else
```

```
    curr2 = curr2->next1;
```

{

```
}
```

```
int main (void)
{
    int a, b, n;
    cin >> a >> b >> n;
```

```
Node *head1 = makeL(a, n);
```

```
int c = gcd(a, b);
```

```
if (c == 1)
```

```
{ c = a * b; }
```

```
print( head1 );
cout << endl;
Node * head2 = makeC( b, n, c, read1 );
print2[ head2 ];
cout << endl;
merge( head1, head2 );
return 0;
```

3