Smallest of three integers without comparison operators

Write a C program to find the smallest of three integers, without using any of the comparison operators.

Let 3 input numbers be x, y and z.

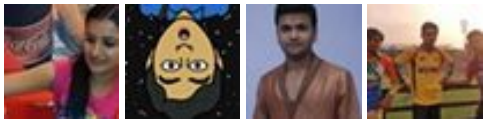
Method 1 (Repeated Subtraction)
Take a counter variable c and initialize it with 0. In a loop, repeatedly subtract x, y and z by 1 and increment c. The number which becomes 0 first is the smallest. After the loop terminates, c will hold the minimum of 3.

```
#include<stdio.h>

int smallest(int x, int y, int z)
{
  int c = 0;
  while ( x && y && z )
  {
      x--;  y--; z--; c++;
  }
  return c;
}

int main()
{
   int x = 12, y = 15, z = 5;
   printf("Minimum of 3 numbers is %d", smallest(x, y, z));
   return 0;
}
```

This methid doesn't work for negative numbers. Method 2 works for negative nnumbers also.



Method 2 (Use Bit Operations)
Use method 2 of this post to find minimum of two numbers (We can't use Method 1 as Method 1 uses comparison operator). Once we have functionality to find minimum of 2 numbers, we can use this to find minimum of 3 numbers.

```
// See mthod 2 of http://www.geeksforgeeks.org/archives/2643
#include<stdio.h>
#define CHAR_BIT 8

/*Function to find minimum of x and y*/
int min(int x, int y)
{
  return  y + ((x - y) & ((x - y) >>
            (sizeof(int) * CHAR_BIT - 1)));
}
```

```
/* Function to find minimum of 3 numbers x, y and z*/
int smallest(int x, int y, int z)
{
    return min(x, min(y, z));
}

int main()
{
   int x = 12, y = 15, z = 5;
   printf("Minimum of 3 numbers is %d", smallest(x, y, z));
   return 0;
}
```

Method 3 (Use Division operator)
We can also use division operator to find minimum of two numbers. If value of (a/b) is zero, then b is greater than a, else a is greater. Thanks to gopinath and Vignesh for suggesting this method.

```
#include <stdio.h>

// Using division operator to find minimum of three numbers
int smallest(int x, int y, int z)
{
    if (!(y/x))  // Same as "if (y < x)"
        return (!(y/z))? y : z;
    return (!(x/z))? x : z;
}
```


Popular Posts

```
int main()
{
    int x = 78, y = 88, z = 68;
    printf("Minimum of 3 numbers is %d", smallest(x, y, z));
    return 0;
}
```

Please write comments if you find the above codes/algorithms incorrect, or find other ways to solve the same problem.

Related Tpoics:

- Check if a number is multiple of 9 using bitwise operators
- How to swap two numbers without using a temporary variable?
- Divide and Conquer | Set 4 (Karatsuba algorithm for fast multiplication)
- Find position of the only set bit
- Swap all odd and even bits
- Add two bit strings
- Write your own strcmp that ignores cases
- Binary representation of a given number

Writing code in comment? Please use **ideone.com** and share the link here.

30 Comments GeeksforGeeks

Sort by Newest

Join the discussion…

ruchi • 2 months ago

can you plz tell me the minimum number of comparisons required to find the l among a set of 4 integers

Reply • Share ›

agnostic ruchi • 2 months ago

we have to do at least 3 comparisons

Reply • Share ›

rihansh • 5 months ago

```
if(a/b&&c/b)
return b;
else if(a/c&&b/c)
return c;
else
return a;
```

Reply • Share ›

Umang Mahajan • 5 months ago

Recent Comments

Abhi You live US or India?

Google (Mountain View) interview · 23 minutes ago

Aman Hi, Why arent we checking for conditions...

Write a C program to Delete a Tree. · 1 hour ago

kzs please provide solution for the problem...

Backtracking | Set 2 (Rat in a Maze) · 1 hour ago

Sanjay Agarwal bool tree::Root_to_leaf_path_given_sum(tree...

Root to leaf path sum equal to a given number · 1 hour ago

GOPI GOPINATH @admin Highlight this sentence "We can easily...

Count trailing zeroes in factorial of a number · 1 hour ago

newCoder3006 If the array contains negative numbers also. We...

Find subarray with given sum · 1 hour ago

int min=(a/b)?((b/c)?c:b):((a/c)?c:a);

This works for positive numbers only. Can some please suggest a similar solu

• Reply • Share ›

Sagar • 7 months ago

Can anyone explain me the working of #method 2 for x=12,y=15,z=5;
Correct me if i go wrong
during the first call to min with values 15,5
x-y=10 and (x-y)>>31 {for a 32 bit compiler}=0
so, (x-y)&((x-y)>>31)=0
and the function returns y i.e 15 instead it should return 5.
Please help me resolve this issue.

• Reply • Share ›

Sunny • 8 months ago

```
int min(int a, int b)
{
return (a-b) & (1<<31) ? a : b;
}
```

• Reply • Share ›

Azim • 8 months ago

Method 3 works only for positive nos.

2 • Reply • Share ›

Vivek Verma • 9 months ago

I guess the Method 3 would also not work for negative numbers. Assume x=-3
But Method 3 would return -3.

• Reply • Share ›

gopinath • a year ago

//one can use ternary operator also in place of if else.

```
#include
int smallest(int a, int b, int c)
{
if(a/b)
{
if(b/c)
return c;
else
return b;
}
else if(a/c)
{
return c;
}
else
return a;
}
main()
{
printf("%d\n",smallest(12,2,56));
}
```

 • Reply • Share ›

Ramasubramani • a year ago

```
int firstMin = (x>y)*y+ (y>x)*x;
int finalMin = (firstMin>z)*z + (z>firstMin)*firstMin;
```

finalMin is the minimum value among the three numbers.

```
/* Paste your code here (You may delete these lines if not writing c
```

 • Reply • Share ›

Kartik ➦ Ramasubramani • a year ago

You have used comparison operator :)

 • Reply • Share ›

unnykrishnan • 2 years ago

The first method will work if we set c=INT_MIN

 • Reply • Share ›

dharmendra • 2 years ago

method 1 is wrong for input of negative no
12 -15 5
o/p 5 is smallest

 • Reply • Share ›

Mahendra Sengar • 2 years ago

/* Paste your code here (You may delete these lines if not writing code) */

#include

using namespace std;

int main()

{

int a = 8;

int b = 10;

```
int diff,msb,max;

diff = a - b;

msb = (diff >> 31) & 0x1;

max = a - msb * diff;

cout<<"Maximum of two numbers is : "<<max<<endl;

return 0;

}
```

^ | v • Reply • Share ›

Mahendra Sengar ➦ Mahendra Sengar • 2 years ago

```
/* Paste your code here (You may delete these lines if not writing code
this is just the trick i wanted to convey ,same approach can be implem
```

^ | v • Reply • Share ›

prashant • 2 years ago

```
# include
void main()
{ int a=5,b=790,c=20;
if(a/b)
{ if(b/c)
printf(" c is min");
else
printf(" b is min");
}
```

```
else
{ if(a/c)
printf(" c is min");
else
printf(" a is min");
}
return 0;
}
```

Reply • Share ›

prashant ➔ prashant • 2 years ago

this is just a substitute for method 1

Reply • Share ›

shaan7 • 2 years ago

Method1 has a while loop, which can't work without comparison. So, it disqual

Reply • Share ›

amitcm • 2 years ago

Earliest occurrence of the most significant bit [after 1st one in signed int, howe
this fact, one can solve the problem. Ignore the tie, if any.

```
/* Paste your code here (You may delete these lines if not writing c
```

Reply • Share ›

Mike • 2 years ago

Mask out the negative bit:

```
int min(int x, int y, int z)
{
    const int mask = 1 << ((sizeof(int) * 4) - 1);
```

```
    if ((x - y) & mask)
        return ((x - z) & mask) ? x : z;
    else
        return ((y - z) & mask) ? y : z;
}
```

• Reply • Share ›

Mike ➦ Mike • 2 years ago

Typo: That 4 should be an 8 (8 bits in a byte)

• Reply • Share ›

Joonas • 2 years ago

Arithmetic/logic unit in the processor has built-in comparison circuits for quick
want to avoid the quickest hardware based solution?

• Reply • Share ›

shaan7 ➦ Joonas • 2 years ago

Thats the problem I find everywhere, companies ask questions in inter

```
/* Paste your code here (You may delete these lines if not wri
```

• Reply • Share ›

Yasir • 2 years ago

```
 int x;  // we want to find the minimum of x and y
int y;
int r;  // the result goes here

r = y ^ ((x ^ y) & -(x < y)); // min(x, y)
```

Reply • Share ›

ramesh Yasir • 2 years ago

u have used comparison operator...

```
/* Paste your code here (You may delete these lines if not wri
```

Reply • Share ›

tvs • 2 years ago

this program doesn't work for negative numbers.

```
/* Paste your code here (You may delete these lines if not writing c
```

Reply • Share ›

kartik tvs • 2 years ago

Method 1 doesn't work for negative numbers, but method 2 works fine.

Reply • Share ›

unnykrishnan kartik • 2 years ago

Method 1 will work for negative numbers if we set c=INT_MIN

Reply • Share ›

bala • 2 years ago

If we sort all the numbers, the very purpose of this question is lost i.e. not to us
sorting can be done without the help of the comparison operators. So sorting i

Reply • Share ›

anmol • 2 years ago

Similarly, minimum of n numbers can be calulated without using any comparis
using same technique.