CS19003: Programming and Data Structures Laboratory

Aritra Hazra,
CSE Dept., IIT Kharagpur

http://cse.iitkgp.ac.in/~aritrah/course/lab/PDS/Spring2021/

20-Apr-2021

Iterative execution (Loops)

The while loop

```
while (condition)
{
   execute loop body;
}
```

GCD by repeated division

```
while (b > 0)
{
   r = a % b;
   a = b;
   b = r;
}
printf("gcd = %d\n",a);
```

Iterative execution (Loops)

The for loop

```
for ( initialize; condition; increment )
{
   execute loop body;
}
```

N^{th} harmonic number $H(n) = \frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{n}$

```
H = 0;
for (i=1; i<=n; ++i) H += 1.0/i;
   printf("H(%d) = %f\n", n, H);
```

Iterative execution (Loops)

The Fibonacci numbers

$F_n = F_{n-1} + F_{n-2}$ for $n \geq 2, F_1 = 1, F_0 = 0$

While

```
i = 1, F = 1; p1 = 0;
while (i < n)
{
  ++i;
  p2 = p1;
  p1 = F;
  F = p1 + p2;
}
```

For

```
p1 = 0, F = 1;
for(i = 2;i <= n;++i)
{
  p2 = p1;
  p1 = F;
  F = p1 + p2;
}
```

```
printf("F(%d)=%d",n,F); //for both programs
```

Loop flow control

- A loop may be *conditionally* broken from inside

```
while (1)
{
 if (b == 0) break;
 r = a % b;
 a = b;
 b = r;
}
printf("gcd = %d\n", a);
```

Loop flow control

- A loop iteration may be *conditionally* skipped
- Ex: Printing 1,2,...,100 neatly with 10 integers per line

```
for (i=1; i<=100; ++i) {
    printf("%4d",i);
    if (i%10 != 0) continue;
    printf("\n");
}
```

Debugging you program: removing logical errors

- First look at your program and see if you can find some obvious logical errors. If found, correct and retry
- If it is not immediately evident, take some (small) input, work out by hand what the values of your variables should be after each step logically
- Put printf statements at those steps and find the first step the program prints a wrong value. Keep repeating until all mistakes are corrected

Bug Localization


```
void main()
{
 int k = 2, n = 1;
 while (k < 7) {
  n = n*k;
  k++;
 }
 printf("After loop 1\n"); /*printf for debugging*/
 while (k != 21) {
  n = n + k;
  k = k+2;
 }/* do not miss \n in debug printf */
 printf("After loop 2\n"); /*printf for debugging*/
 printf("n is %d\n", n);
}
```

- Program hangs, second loop does not terminate
- Statement "After loop 2" is not printed, So you know the first loop finished and the second did not.

GCD: erroneous implementations

Correct

```
while (1)
{
 if (b == 0) break ;
 r = a % b;
 a = b;
 b = r;
}
```

always o/p = 0

```
while (1)
{
 if (b == 0) break ;
 r = a % b;
 b = r;
 a = b;
}
```

```
/*last 2 statements exchanged*/
```

o/p is in 'a'. In R.H.S program, a=0 due to the chaining effect when 'r' is 0

Debugging a single block

Executing the correct program with a=45, b=12

```
 while (1)
 {
  if (b==0) break ;
  r = a % b;                    /* iter 1 values*/
  printf("a=%d,b=%d,r=%d\n"); /* 45,12,9*/
  a = b;
  printf("a=%d,b=%d,r=%d\n"); /* 12,12,9*/
  b = r;
  printf("a=%d,b=%d,r=%d\n"); /* 12, 9,9*/
 }
printf ("gcd = %d\n", a);
```

We expect a = old value of b = 12 , b = r = a%b =9
so, this is fine

Debugging a single block

Executing the incorrect program with a=45, b=12

```
while (1)
{
 if (b==0) break ;
 r = a % b;                    /* iter 1 values*/
 printf("a=%d,b=%d,r=%d\n"); /* 45,12,9*/
 b = r;
 printf("a=%d,b=%d,r=%d\n"); /* 45, 9,9*/
 a = b;
 printf("a=%d,b=%d,r=%d\n"); /*  9, 9,9*/
}
printf ("gcd = %d\n", a);
```

We expect a = old value of b = 12 , b = r = a%b =9

Only r is assigned correctly, problem with code after r=a%b

GCD: some more erroneous implementations :)

Infinite loop

```
while (1)
{
 if (b == 0) break ;
 r = a % b;
 a = b;
 b = a;
}
/*b=a by mistake*/
```

Divide by zero

```
 while (1)
{
 if (a == 0) break ;
 r = a % b;
 a = b;
 b = r;
}
/*a==0 by mistake*/
```

Nested loop:

```
int i, j;
/* print header line: */
printf("   ");
for(j = 1; j <= 10; j = j + 1)
  printf(" %3d", j);
printf("\n");
/* print table: */
for(i = 1; i <= 10; i = i + 1)
{
  printf("%2d", i);
  for(j = 1; j <= 10; j = j + 1)
    printf(" %3d", i + j);
  printf("\n");
}
return 0;
```

Output table

	1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10	11
2	3	4	5	6	7	8	9	10	11	12
3	4	5	6	7	8	9	10	11	12	13
4	5	6	7	8	9	10	11	12	13	14
5	6	7	8	9	10	11	12	13	14	15
6	7	8	9	10	11	12	13	14	15	16
7	8	9	10	11	12	13	14	15	16	17
8	9	10	11	12	13	14	15	16	17	18
9	10	11	12	13	14	15	16	17	18	19
10	11	12	13	14	15	16	17	18	19	20

Make a simple modification to the program to print a multiplication table, or a subtraction table

break statement in loop nest

```
void main()
{
   int low, high, desired, i, flag = 0;
   scanf("%d %d %d", &low, &high,&desired);
   i = low;
   while (i < high) {
      for (j = i+1; j <= high; ++j) {
          if (j % i == desired) {
              flag = 1;
              break; //breaks from for loop
          }
      }
      if (flag == 1) break;
      i = i + 1; //breaks from while loop
   }
}
```

Thank You