Beginner’s Questions

Susan Eisenbach

21 November 2007

Write a program that reads a list of positive numbers < 100 and displays the largest and the smallest. A 0 (the number zero) should terminate the list. Do not use an array to answer this question. Do not use any methods either. There is at least one number before the 0.

If one looks at the square numbers and the differences between each adjacent pair of square numbers a pattern emerges.

squares: 0 1 4 9 16 25 36 49…

differences: 1 3 5 7 9 11 13…

Write a program to print out the square numbers, ten to a line, *without* using multiplication `*`.

b i) Write a method called `absDiff`, which takes two integers as parameters and returns the absolute value of the difference between the two integers.

ii) Write a method called `intRoot`, which takes an integer as a parameter and returns the integer value closest to the square root of the parameter. Your method may call `absDiff`.

iii) Given a sequence where the i^{th} term is the closest integer to the square root of i, write a method called `printSequence`, which takes an integer `n` as a parameter and prints on the screen the first n integers in the sequence, ten to a line. You may use previously written and inbuilt methods. So `printSequence(10)` would print on the screen:

```
1     1     2     2     2     2     3     3     3     3
```

Write a method `numFound` that takes two integers `n` and `m` and returns the number of occurrences of `m` in the sequence of the first `n` integer square roots. So

`numFound(10,3)` is `4`

`numFound(10,1)` is `2`

`numFound(10,5)` is `0`

You may use previously written and inbuilt methods.

1

```
void main(){
int diff;
int square;
int i;
  diff = 1;
  square = 0;
  for (i = 1; i<=100; i++){
    print(square +" ");
    square = square + diff;
    diff = diff + 2;
    if (i % 10 == 0)  {println(" ");}
  }
}
```

2i)

```
int absDiff(int a, int b){
      if (a<b) {return b-a;}
      else {return a-b;}
}
```

2ii)

```
int intRoot(int n){
      int diff = n;
      int root = n;
      int d = n;
      int i;
      for (i = 1; i <= n; i++){
            d = absDiff(i*i,n);
            if (d < diff){diff = d; root = i;}
      }
      return root;
}
```

2iii)

```
void printSequence(int n){
int i;
  for (i = 1; i <= n; i++){
    print(intRoot(i)+"  ");
    if (i % 10 == 0) {println(" ");}
  }
}



void main(){
  print("Please type in your number -->");
    printSequence(readInt());
}

int absDiff(int a, int b){
  if (a<b) {return b-a;}
  else {return a-b;}
}

int intRoot(int n){
int diff = n;
int root = n;
int d = n;
int i;
```

```
  for (i = 1; i <= n; i++){
        d = absDiff(i*i,n);
        if (d < diff){diff = d; root = i;}
  }
  return root;
}

void printSequence(int n){
int i;
  for (i = 1; i <= n; i++){
    print(intRoot(i)+"  ");
    if (i % 10 == 0) {println(" ");}
  }
}
```

2v)

```
int numFound(int n, int m){
int f = 0;
int i;
   for (i = 1; i <= n; i++){
     if (intRoot(i) == m) {f = f+1;}
   }
   return f;
}
```