Xsquare and Two Arrays - Hackerearth

Tuesday, January 14, 2020 8:53 PM

The Code:

```
#include <stdio.h>

int main(){
   int length, queryCount, iter, left, right, type, scaledLeftBoundary;
   scanf("%d %d", &length, &queryCount);

   long long int *A, *B;
   long long int ASum, BSum, totalSum;
   A = (long long int*)calloc(length+1, sizeof(long long int));
   B = (long long int*)calloc(length+1, sizeof(long long int));

   A[0] = 0;
   B[0] = 0;

   for(iter=1; iter<=length; iter++){
      scanf("%lld", &A[iter]);

      if(iter-2 >= 0){
         A[iter] = A[iter] + A[iter-2] ;
      }
   }

   for(iter=1; iter<=length; iter++){
      scanf("%lld", &B[iter]);

      if(iter-2 >= 0){
         B[iter] = B[iter] + B[iter-2] ;
      }
   }

   for(iter=0; iter<queryCount; iter++){

      scanf("%d %d %d", &type, &left, &right);

      scaledLeftBoundary = left-2 >= 0 ? left-2 : 0;
      if(type == 1){
         if((left % 2 == 0 && right % 2 == 0) || (left % 2 != 0 && right % 2 != 0)){
            ASum = A[right] - A[scaledLeftBoundary];
            BSum = B[right-1] - B[left-1];
         }
         else{
            ASum = A[right-1] - A[scaledLeftBoundary];
            BSum = B[right] - B[left-1];
         }
      }
      else{
```

```
        if((left % 2 == 0 && right % 2 == 0) || (left % 2 != 0 && right % 2 != 0)){
            BSum = B[right] - B[scaledLeftBoundary];
            ASum = A[right-1] - A[left-1];
        }
        else{
            BSum = B[right-1] - B[scaledLeftBoundary];
            ASum = A[right] - A[left-1];
        }
      }

      totalSum = ASum + BSum;

      printf("%lld\n", totalSum);
   }

}
```

The Stats:

Time (sec)
0.107011

Memory (KiB)
64

Language
C