Submission

ID	DATE	PROBLEM	STATUS	CPU	LANG
	TEST CASES				
4790450	12:11:50	Coloring Socks	✓ **Accepted**	0.08 s	Python 3
	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓				

Submission contains 1 file: download zip archive

FILENAME	FILESIZE	SHA-1 SUM	
color_260621270.py	788 bytes	c21ab7f4122b86b78bbb71bd5ac2dfede9209637	download

Edit and resubmit this submission.

color_260621270.py

```
# split takes the list of numbers and the 2nd and 3rd inputs
# the first while loop acts as a for loop for arrays as python only has
# for each loops. then because the list gets sorted, one can start at
# the very end and compare that to the max going down

def split(numbers, c, k):
  s = len(numbers)
  washes = 0
  load = 1
  lower = 0
  numbers = sorted(numbers)

  if(c >= s): return 1
  while lower < len(numbers):
    upper = lower + c - 1
    if upper >= s: upper = s - 1
    for x in numbers:
      if numbers[upper] - numbers[lower] <= k:
        washes += 1
        lower = upper + 1
        break
      upper -= 1
  return washes

variables = input().split()
```

```
c = int(variables[1])
k = int(variables[2])
sock_diff = input().split()
print(split(list(map(int, sock_diff)), c, k))
```

Submission

ID	DATE	PROBLEM	STATUS	CPU	LANG
	TEST CASES				
4790482	12:15:42	Flexible Spaces	✓ **Accepted**	0.02 s	Python 3
	✓✓✓✓✓✓✓✓✓✓✓✓✓✓				

Submission contains 1 file: download zip archive

FILENAME	FILESIZE	SHA-1 SUM	
flexible_260621270.py	648 bytes	`9d58c521fb573cc4fc994c3bfaff75d1e54768f1`	download

Edit and resubmit this submission.

flexible_260621270.py

```python
# split takes the list of numbers and the 2nd input, the width
# the program proceeds to take the difference between every single element
# and every other element. the result will then have the 0 removed,
# sorted and printed

def split(numbers, w):
  s = len(numbers)
  numbers.append(0)
  numbers.append(w)
  numbers = sorted(numbers)
  lower = 0
  mylist =[]
  for x in numbers:
    for y in numbers:
      mylist.append(abs(y-x))
  mylist = list(dict.fromkeys(mylist))
  mylist.remove(0)
  return sorted(mylist)

variables = input().split()
w = int(variables[0])
num_line = input().split()
print(*split(list(map(int, num_line)), w), sep = ' ')
```

Submission

ID	DATE	PROBLEM	STATUS	CPU	LANG
	TEST CASES				
4790478	12:15:07	Radio Commercials	✔ **Accepted**	0.07 s	Python 3
	✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔				

Submission contains 1 file: download zip archive

FILENAME	FILESIZE	SHA-1 SUM	
commercials_260621270.py	568 bytes	ed2b4f0f563014a496be0f24fdb84e3e387e048f	download

Edit and resubmit this submission.

commercials_260621270.py

```python
# https://en.wikipedia.org/wiki/Maximum_subarray_problem
# "We can compute the maximum subarray sum ending at position i for all positions i
# by iterating once over the array. As we go, we simply keep track of the maximum sum
# we've ever seen."

def max_subarray(numbers):
  best_sum = 0  # or: float('-inf')
  current_sum = 0
  for x in numbers:
    current_sum = max(0, current_sum + x)
    best_sum = max(best_sum, current_sum)
  return best_sum

p = int((input()).split()[1])
profit = list(map(lambda z: int(z) - p, input().split()))
print(max_subarray(profit))
```