TASK 12: elementary data structure algorithms-based questions and finding complexity.

Little Shino and Pairs:

Given a permutation of numbers from 1 to N. among all the subarrays, find the number of unique pair such that a and a is maximum and b is second maximum in the array.

Input:

first line contains an integer, N second line contains N space seperated distinct integers, denoting the permutation.

Output:

print the required answer

SAMPLE INPUT:

5

1 2 3 4 5

SAMPLE OUTPUT: 4

Sure.

algorithm:

1. Read the input values of N and the permutation array A.
2. Initialize a variable count to 0, keep track of the number of valid pairs.
3. Loop over all possible subarrays of A with nested loops i and j.

a) Find the maximum element, max1, and its index, max1idx in the subarray [i,j].

4. Print the final value of count.

program:

```
#include <stdio.h>
int main(){
  int n, i, j, max, second_max, count=0;
  scanf("%d", &n);
  int arr[n];
for(i=0; i<n; i++){
  max = arr[i];
  second_max = -1;
   if(array[i] > max){
  second_max = max;
  max = arr[i];
}
```

```
else if (arr[i] > second_max){
        second_max = arr[i];
}
if (second_max != -1 && arr[i] == second_max){
count++;
break;
}
}
}
printf("%d", count);
return 0;
}
```

output:

SAMPLE INPUT: 5

1 2 3 4 5

SAMPLE OUTPUT: 4

algorithm:

1. Calculate the sum of first n natural numbers.
2. create an array of size equal to the sum calculated in step 1.
3. Loop through the range from 1 to n+1.
4. return the variable result as the output.

Program:

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    int T, n, i, j, k, sum, size;
scanf("%d", &T);
while(T--){
scanf("%d", &n);
size = (n*(n+1))/2;
int *arr = (int*) malloc (sizeof(int)*size);
for(i=0; i<size; i++){
 if(i%2!=0){
for(j=0; j<i; j++){
   int max = j;
for(int k=j+1; k<i; k++){
if(arr[k] > arr[max]){
max = k;
}
}
int temp = arr[j];
arr[j] = arr[max];
}
max = 1;
}
```

```
}
int temp = arr[j+k];
arr[j+k] = arr[max+k]; arr[max+k] = temp;
}
sum += arr[k]; k++;
}
else{
for(j=0; j<i; j++){
int min = j
for(int i=j+1; i<i; i++){
if(arr[i+k]<arr[min+k]){
min = i;
}
}
}
sum += arr[k]; k += 1;
}
}
printf("%d\n", sum); free(arr);
}
return 0;
}
```

Output:

Input: 1

3

1 2 3 4 5 6

Output:

9.

| VEL TECH - CSE | |
|---|---|
| EX NO. | 1 |
| PERFORMANCE (5) | 5 |
| RESULT AND ANALYSIS (3) | 3 |
| VIVA VOCE (3) | 3 |
| RECORD (4) | 4 |
| TOTAL (15) | 15 |
| SIGN WITH DATE | |

Result

Thus the program is executed and verified successfully.