DAY 1 – DS Using C Programs

```c
/*
You are given an array of prices where prices[i] is the price of a given stock
on an ith day.

You want to maximize your profit by choosing a single day to buy one stock and
choosing a different day in the future to sell that stock.

Return the maximum profit you can achieve from this transaction. If you cannot
achieve any profit, return 0.

Example 1:
Input: prices = [7,1,5,3,6,4]
Output: 5

Explanation: Buy on day 2 (price = 1) and sell on day 5 (price = 6), profit =
6-1 = 5.
Note that buying on day 2 and selling on day 1 is not allowed because you must
buy before you sell.

Example 2:
Input: prices = [7,6,4,3,1]
Output: 0

Explanation: In this case, no transactions are done, and the max profit = 0.
*/

#include <stdio.h>
#include <stdlib.h>

int maxProfit(int prices[], int n) {
    int profit, max, total_max = 0, day, stock;

    for (day = 1; day <= n; day++)
    {
        max = 0;
        for (stock = day + 1; stock <= n; stock++)
        {
            profit = prices[stock] - prices[day];
            if (profit > max)
            {
                max = profit;
            }
        }
        if (max > total_max)
        {
            total_max = max;
```

```
        }
    }
    if (total_max > 0)
        return total_max;
    else
        return 0;
}

void main()
{
    int prices[100], n, i, total_max;

    printf("\nEnter the no. of array prices[] elements: ");
    scanf("%d", &n);

    for (i = 1; i <= n; i++)
    {
        printf("\nprices[%d] = ", i);
        scanf("%d", &prices[i]);
    }

    total_max = maxProfit(prices, n);

    if (total_max > 0)
        printf("\nGREAT :)!! Your Maximum Profit is = %d", total_max);
    else
        printf("\nOOPS :(!! Your Maximum Profit is = %d", total_max);

    printf("\n");
}
```

OUTPUT:

```
Enter the no. of array prices[] elements: 6

prices[1] = 7

prices[2] = 1

prices[3] = 5

prices[4] = 3

prices[5] = 6

prices[6] = 4

GREAT :)!! Your Maximum Profit is = 5
```

```
Enter the no. of array prices[] elements: 5

prices[1] = 7

prices[2] = 6

prices[3] = 4

prices[4] = 3

prices[5] = 1

OOPS :(!! Your Maximum Profit is = 0
```