Submission by : Harshal Nandigramwar
Email: pro.bbcom18@gmail.com

1. **Task Statement**

a. Write a program in C language. Print even and odd characters of a string using function. You must use flag 0 and flag 1 to display.

Input: stringprint(“FEEDBYME”, 0)
Output: E D Y E

Input: stringprint(“FEEDBYME”, 1)
Output: F E B M

- **Solution**

Time complexity: O(|s|)
Space complexity: O(1)

```c
#include <stdio.h>   // IO
#include <string.h>  // strlen()

void stringprint(char* string, int flag) {
   size_t sz = strlen(string);

   // If flag is 0 we print odd positions.
   // If flag is 1 we print even positions.
   // Makes flag as the start index.
   // If the flag is 1, the start index is 0.
   // If the flag is 0, the start index is 1.
   flag = 1 - flag;

   // Loops through all alternate characters
   // starting from the flag.
   for (size_t i = flag; i < sz; i += 2) {
       printf("%c ", string[i]);
   }

   // Formatting of output(Optional).
   printf("\n");
```

```c
}

int main() {
   char string[1000];  // Input string.
   int flag;           // Input flag.

   printf("Enter a string:");
   scanf("%s", string);

   printf("Enter a flag value(0/1):");
   scanf("%1d", &flag);

   if (flag > 1) {
       printf("Invalid input\n");
       return 0;
   }

   stringprint(string, flag);

   return 0;
}
```

b. Write a program in C language. Given a list containing future predictions of share prices, find the maximum profit that can be earned by buying and selling shares any number of times with constraint that a new transaction can only start after the previous transaction is complete. i.e . we can only hold at-most one share at a time.

Input (Stock prices):
{100, 108, 260, 310, 40, 535, 696}.

Output:
Buy on - 5, sell on - 7, Profit = 655

Note: The given sample is contradictory to the statement. According to the statement maximum profit can be 866 (Buy on 1, Sell on 4, Buy on 5, Sell on 7). If only one transaction is allowed then the given sample may be applicable but that is not the case. The solution below follows the statement given.

- **Solution**

  Time complexity: O(N) N is the size of the list.
  Space complexity: O(1)

```c
#include <stdio.h>

void findTransactions(const int *list, int noOfDays){

   int profit = 0, lastBought = 0;

   for (int i=1; i<noOfDays; i++){
       if (list[i] < list[i - 1]){
           profit += list[i - 1] - list[lastBought];

           if (lastBought != i - 1){
               printf("Buy on - %d\nSell on - %d\n\n", lastBought + 1, i);
           }

           lastBought = i;
       }
   }

   profit += list[noOfDays - 1] - list[lastBought];

   if (lastBought != noOfDays - 1){
       printf("Buy on - %d\nSell on - %d\n\n", lastBought + 1, noOfDays);
   }

   printf("Max Profit: %d\n", profit);
}

int main(){

   int noOfDays;

   printf("Enter no. of days:");
```

```
    scanf("%d", &noOfDays);

    int list[noOfDays];

    printf("Enter prices (space separated):");
    for(int i=0; i<noOfDays; i++){
        scanf("%d", &list[i]);
    }

    findTransactions(list, noOfDays);

    return 0;
}
```