
FACE

TCS - Digital

Programs

FACE

Question 1

Given a value N, find the number of ways to make change for N cents, if we have infinite supply of each of S = { S1, S2, .. , Sm} valued coins. The order of coins doesn't matter.

Sample Input:

3
1 2 3
4

Sample Output:

4

FACE

```c
#include<stdio.h>

int main()
{
      int test_case,size,i,val;
      scanf("%d",&size);
      int arr[size];
      for(i=0;i<size;i++)
        scanf("%d",&arr[i]);
      scanf("%d",&val);
      printf("%d",count(arr,size,val));
      return 0;
}
```

```
1
2   int count( int S[], int m, int n )
3   {
4       int i, j, x, y;
5       int table[n + 1][m];
6       for (i = 0; i < m; i++)
7           table[0][i] = 1;
8       for (i = 1; i < n + 1; i++)
9       {
10          for (j = 0; j < m; j++)
11          {
12              x = (i-S[j] >= 0) ? table[i - S[j]][j] : 0;
13              y = (j >= 1) ? table[i][j - 1] : 0;
14              table[i][j] = x + y;
15          }
16      }
17      return table[n][m - 1];
18  }
19
20
21
22
```

OUTPUT

```
4
2 5 3 6
10
```

Question 2

Suppose you have N eggs and you want to determine from which floor in a K-floor building you can drop an egg such that it doesn't break. You have to determine the minimum number of attempts you need in order find the critical floor in the worst case while using the best strategy. There are few rules given below.

- An egg that survives a fall can be used again.
- A broken egg must be discarded.
- The effect of a fall is the same for all eggs.
- If the egg doesn't break at a certain floor, it will not break at any floor below.
- If the eggs breaks at a certain floor, it will break at any floor above.

Question 2

Sample Input:

2
10

Sample Output:

4

```
1   #include<stdio.h>
2   #include<limits.h>
3   int max(int a,int b){
4    return a>b?a:b;
5   }
6   int func(int n,int k)
7   {
8       int arr[n+1][k+1];
9       int i,j,x,res;
10      for(i=0;i<=n;i++)
11      arr[i][0]=0;
12       for(i=0;i<=n;i++)
13         arr[i][1]=1;
14         for(j=1;j<=k;j++)
15         {
16             arr[1][j]=j;
17         }
18
19
20
21
22
```

```
1          for(i=2;i<=n;i++){
2              for(j=2;j<=k;j++){
3                  arr[i][j]=INT_MAX;
4                  for(x=1;x<=j;x++){
5                    res=1+max(arr[i-1][x-1],arr[i][j-x]);
6                    if(res<arr[i][j]){
7                        arr[i][j]=res;
8                    }
9                  }
10             }
11         }
12         return arr[n][k];
13 }
14 int main() {
15
16         int a,b;
17         scanf("%d %d",&a,&b);
18         printf("%d\n",func(a,b));
19         return 0;
20 }
21
22
```

OUTPUT

```
2
10

4
```

Question 3

Gayathri is interested to solve puzzle questions and her brother gave a puzzle to solve. In the puzzle, he gave an array consisting of zeros and ones, you are allowed to flip at most 1 element from 0 to 1. Find the size of the sub-array which consists of the maximum number of consecutive 1's . Help her by writing a program.

Sample Input:

```
10
1 0 1 1 1 0 1 1 1 0
```

Sample Output:

```
7
```

FACE

```c
#include<stdio.h>

int get_max_consecutive1(int arr[], int size);

int max_element(int x, int y);

int main()
{
  int arr[100];
  int i, N;
  scanf("%d", &N);
  for(i = 0; i < N; i++)
      scanf("%d", &arr[i]);

  get_max_consecutive1(arr, N);
  return 0;
}
```

```
1  int max_element(int x, int y){
2      if(x >= y){
3          return x;
4      }
5      return y;
6  }
7  int get_max_consecutive1(int arr[], int N){
8    int i, count = 0, flip_count = 0, max_count = 0;
9    for(i = 0; i < N; i++){
10     if(arr[i] == 1){
11         count++;
12     }
13     if(arr[i] == 0){
14         if(arr[i+1] == 0){
15             flip_count = 1;
16         }
17         else{
18             flip_count = count + 1;
19             count = 0;
20         }
21      }
22
```

```
        max_count = max_element(count + flip_count, max_count);
    }
    printf("%d", max_count);
}
```

OUTPUT

```
10
1 0 1 1 1 0 1 1 1 0

7
```

Question 4

Write a program to find whether the given string is rotated or not

Sample Input:

abcd
bcda

Sample Output:

Rotated

```c
# include <stdio.h>
# include <string.h>
int are_rotated(char str1[], char str2[]) ;
int main()
{
    char str1[100],str2[100];
    gets(str1);
    gets(str2);
    if (are_rotated(str1, str2))
       printf("Rotated");
    else
       printf("Not rotated");
    return 0;
}
```

```
1  int are_rotated(char str1[], char str2[])
2  {
3    int size1   = strlen(str1);
4    int size2   = strlen(str2);
5    int i;
6    char temp[100];
7    void *ptr;
8    if (size1 != size2)
9       return 0;
10   temp[0] = ' ';
11
12   strcat(temp, str1);
13   strcat(temp, str1);
14   ptr = strstr(temp, str2);
15   if (ptr != NULL)
16     return 1;
17   else
18     return 0;
19 }
20
21
22
```

OUTPUT

```
where
reehw

Not rotated
```

Question 5

You have 100 cards, numbered 1 to 100. You distribute them into k piles and collect back the piles in order.
For example, if you distribute them into 4 piles, then the first pile will contain the cards numbered 1, 5, 9, ... and the 4th pile will contain the cards numbered 4, 8, 12, ... While collecting back the cards you collect first the last pile, flip it bottom to top, then take the third pile, flip it bottom to top and put the cards on top of the 4th pile and so on.

Next round, you distribute the cards into another set of piles and collect in the same manner (last pile first and first pile last).

FACE

Question 5

If we have 10 cards, and put them into 2 piles, the order of the cards in the piles (top to bottom) would be
9, 7, 5, 3, 1 and 10, 8, 6, 4, 2
We flip the piles to get the order
1, 3, 5, 7, 9 and 2, 4, 6, 8, 10
We put second pile at the bottom and first on top of it to get the deck
1, 3, 5, 7, 9, 2, 4, 6, 8, 10

Given the number of rounds (m), number of piles in each round (ki), you need to write a program to find the Nth card from the top at the end of the final round.

FACE

Question 5

Sample Input:

2 2 4

Sample Output:

13

```
1  #include<stdio.h>
2  int main(){
3        int m;
4        scanf("%d", &m);
5        int pile[m];
6        for(int i = 0; i< m; i++)
7              scanf("%d", &pile[i]);
8        int N;
9        scanf("%d", &N);
10       int numbers[100];
11       int num = 1;
12       for(int i = 0; i < 100; i++){
13             numbers[i] = num++;
14       }
15       int arr[100][100];
16       int count = 0;
17       while(count < m){
18             int r = pile[count];
19             int c = 100 / pile[count];
20             if(pile[count] % 2 != 0){
21                   c = c + 1;
22             }
```

```
1           int num = 0;
2               int x = 0;
3               int flag = 0;
4               for(int i = 0; i < r; i++){
5                       num = x;
6                       for(int j = 0; j < c; j++){
7                               if(j == 0){
8                                       arr[i][j] = numbers[num];
9                                       num = num + pile[count];
10                              }
11                              else{
12                                      int val = numbers[num];
13                                      num = num + pile[count];
14                                      if(val > 100){
15                                              break;
16                                      }
17                                      else{
18                                              arr[i][j] = val;
19                                      }
20                              }
21                      }
22                      x++;}
```

```
1               int index = 0;
2               for(int i = 0; i < r; i++){
3                     for(int j = 0; j < c; j++){
4                           if(arr[i][j] != 0){
5                                 numbers[index++] = arr[i][j];
6                           }
7                     }
8               }
9               count++;
10        }
11        printf("%d", numbers[N - 1]);
12  }
13
14
15
16
17
18
19
20
21
22
```

OUTPUT

```
2 2 4

13
```

Question 6

Given an (M X M) Matrix, print the upper triangular values in the diagonal order.

Sample Input:

3 3
1 2 3
4 5 6
7 8 9

Sample Output:

1 5 9 2 6 3

FACE

```c
#include<stdio.h>

int upper_matrix(int r, int c, int arr[r][c]);

int main()
{
  int r, c, i, j;
  scanf("%d%d", &r, &c);
  int matrix[r][c];
  for(i = 0; i < r; i++)
  {
    for(j = 0; j < c; j++)
    {
        scanf("%d", &matrix[i][j]);
    }
  }

  upper_matrix(r, c, matrix);
  return 0;
}
```

```
1  int upper_matrix(int r, int c, int matrix[r][c])
2  {
3    int i, j, k;
4    for(k = 0; k < c; k++)
5    {
6        for(i = 0, j = k; j < c ; i++, j++)
7        printf("%d ", matrix[i][j]);
8    }
9  }
10
11
12
13
14
15
16
17
18
19
20
21
22
```

OUTPUT

```
2 2
1 2
3 4

1 4 2
```

Question 7

Write a program to print all the duplicate characters.

Sample Input:

hello

Sample Output:

l

```c
#include<stdio.h>
int main(){
  char str[100];
  int i, j;
  scanf("%[^\n]s", str);
  for( i = 0; str[i]!='\0'; i++){
    int isDuplicate = 0;
    int matchingIndex;
    for(int j = 0; str[j]!='\0'; j++) {
        if(i == j)
        {
            continue;
        }
            if(str[i] == str[j])
        {
            isDuplicate = 1;
            matchingIndex = j;
            break;
        }
    }


```

```
1       if(isDuplicate == 1 && matchingIndex > i)
2       {
3           printf("%c", str[i]);
4       }
5     }
6     return 0;
7   }
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
```

OUTPUT

```
aaaaaaa

a
```

Question 8

Write a program to find all the consecutive prime sum numbers up to the given range.

Sample Input:

20

Sample Output:

5 17

```
1   #include<stdio.h>
2
3   // Returns 1 if num is prime number
4   // Otherwise returns 0.
5   int isprime(int num)
6   {
7     int i;
8     int isprime = 1;
9     for(i = 2;  i < num; i++)
10    {
11         if(num % i == 0)
12         {
13             isprime = 0;
14             break;
15         }
16     }
17     return isprime;
18  }
19
20
21
22
```

```
1  // Returns 1 if num can be represented as sum of
2  // consecutive prime numbers. Otherwise returns 0.
3  int is_prime_num_has_cons_sum_fact(int num){
4      int prime_num[100], i, j, isFound = 0, sum =0;
5      int prime_num_idx = 0;
6      for(i = 2; i < num; i++){
7          if(isprime(i) == 1){
8              prime_num[prime_num_idx] = i;
9              prime_num_idx++;
10         }
11     }
12     for(i = 0; i < prime_num_idx; i++){
13         sum = 0;
14         for(j = i; j < prime_num_idx; j++)
15         {
16             sum = sum + prime_num[j];
17             if(sum == num){
18                return 1;
19             }
20         }
21     }
22     return 0;}
```

```
1  int main()
2  {
3    int n, m;
4    scanf("%d", &n);
5    for(m=2; m<=n; m++)
6    {
7        if(isprime(m) && is_prime_num_has_cons_sum_fact(m))
8        {
9              printf("%d ", m);
10       }
11   }
12   return 0;
13 }
14
15
16
17
18
19
20
21
22
```

OUTPUT

```
100

5 17 23 31 41 53 59 67 71 83 97
```

Question 9

Write a code to sort the elements of the given array in frequency order.

Sample Input:

10
5 5 5 4 4 3 3 3 3 2

Sample Output:

3 3 3 3 5 5 5 4 4 2

```c
#include<stdio.h>
int main(){
    int n;
    scanf("%d", &n);
    int arr[n];
    for(int i = 0; i < n ; i++){
        scanf("%d", &arr[i]);
    }
    frequncy_sorting(arr, n);
}
```

```
1  void frequncy_sorting(int arr[], int n){
2      int ele_count[n];
3      int element[n];
4      int idx = 0, count = 0;
5      for(int i = 0; i < n - 1; i++){
6          if(arr[i] == arr[i + 1]){
7              count++;
8              if(i + 1 == n - 1){
9                  ele_count[idx] = ++count;
10                 element[idx] = arr[i];
11                 idx++;
12             }
13         }
14         else if(arr[i] != arr[i + 1]){
15             ele_count[idx] = ++count;
16             count = 0;
17             element[idx] = arr[i];
18             idx++;
19             if(i + 1 == n - 1){
20                 ele_count[idx] = ++count;
21                 element[idx] = arr[i + 1];
22                 idx++;
```

```
1               }
2           }
3       }
4       for(int i = 0; i < idx; i++)
5       {
6           for(int j = i + 1; j < idx; j++)
7           {
8               if(ele_count[i] <= ele_count[j]){
9                   int t = ele_count[i];
10                  ele_count[i] = ele_count[j];
11                  ele_count[j] = t;
12                  int x = element[i];
13                  element[i] = element[j];
14                  element[j] = x;
15              }
16          }
17      }
18
19
20
21
22
```

```
1       for(int i = 0; i < idx; i++)
2       {
3           for(int j = i + 1; j < idx; j++)
4           {
5               if(ele_count[i] == ele_count[j]){
6                   int t = ele_count[i];
7                   ele_count[i] = ele_count[j];
8                   ele_count[j] = t;
9                   int x = element[i];
10                  element[i] = element[j];
11                  element[j] = x;
12              }
13          }
14      }
15      for(int i = 0; i < idx; i++){
16          for(int j = 0; j < ele_count[i]; j++){
17              printf("%d ", element[i]);
18          }
19      }
20  }
21
22
```

OUTPUT

```
10
5 5 5 4 4 3 3 3 3 2

3 3 3 3 5 5 5 4 4 2
```

Question 10

Given a string containing alphanumeric characters, calculate the sum of all numbers present in the given string

Sample Input:

1abc23

Sample Output:

24

FACE

```
1   #include<stdio.h>
2   #include<string.h>
3   int main()
4   {
5       char a[20];int sum=0,len,b[20],i,j,k,m,n;
6       scanf("%s",a);
7       len = strlen(a);
8       for(i = 0; i < len; i++)
9       {
10          j = a[i];
11          if(j > 47 && j < 58)
12          {
13              if(a[i+1] > 47 && a[i+1] < 58)
14              {
15                  m = a[i+1];
16                  k = (j - 48) * 10 + (m - 48);
17
18
19
20
21
22
```

```
1                   if(a[i+2] > 47 && a[i+2] < 58)
2                    {
3                        n = a[i+2];
4                            n -= 48;
5                        sum += k * 10 + n;
6                        i += 2;
7                    }
8                    else
9                    {
10                       sum += k;
11                       i += 1;
12                   }
13              }
14              else
15              {
16                  sum += j - 48;
17              }
18          }
19      }
20      printf("%d",sum);
21      return 0;
22  }
```

OUTPUT

```
ggh2bg23v2b1


28
```

Question 11

In a party everyone is in couple except one. People who are in couple have same numbers. Find out the person who is not in couple.

Sample Input:

5
1 2 3 2 1

Sample Output:

3

```
#include<stdio.h>
int main() {
    int n;
      scanf("%d",&n);
      int a[n];
      int ans;
      for(int i=0;i<n;i++){
          scanf("%d",&a[i]);
          ans=a[0];
          for(int i=1;i<n;i++){
              ans=ans^a[i];
          }
      }
      printf("%d",ans);
      return 0;
}
```

OUTPUT

```
5
1 2 1 3 3

2
```

Question 12

Jarvis is weak in computing palindromes for Alphanumeric characters. While Ironman is busy fighting with Thanos, he needs to activate sonic punch but Jarvis is stuck in computing palindromes. You are given a string S containing alphanumeric characters. Find out whether the string is a palindrome or not. If you are unable to solve it then it may result in the death of Iron Man.

Sample Input:

Ab?/Ba

Sample Output:

Yes

FACE

```
1   #include <stdio.h>
2   #include<string.h>
3   #define MAX 100000
4   int is_palindrome(char temp[]);
5   int main(){
6       char str[MAX];
7       scanf("%[^\n]s", str);
8       char temp[MAX];
9       int j = 0, i = 0;
10      for(i = 0; str[i] != '\0'; i++){
11          if((str[i] >= 'a' && str[i] <= 'z') || (str[i] >= 'A' &&
12             str[i] <= 'Z') || (str[i] >= '0' && str[i] <= '9')){
13              if(str[i] >= 'A' && str[i] <= 'Z'){
14                  temp[j++] = str[i] + 32;
15              }
16              else{
17                  temp[j++] = str[i];
18              }
19          }
20      }
21      temp[j] = '\0';
22
```

```
1  if(is_palindrome(temp)){
2          printf("Yes");
3      }
4      else{
5          printf("No");
6      }
7  }
8  int is_palindrome(char temp[]){
9      int len = strlen(temp);
10     int is_found = 1;
11     int s = 0;
12     int e = len -1;
13     while(s <= e){
14         if(temp[s] != temp[e]){
15             is_found = 0;
16             break;
17         }
18         s++;
19         e--;
20     }
21     return is_found;
22 }
```

OUTPUT

```
Do ?972 79 o@f

No
```

Question 13

Everyone has some habits to collect one thing or the other. Harshit also has the craze to collect pens but in 3 piles.
In the first pile, he collected A pens and in the second pile, he collected B pens but in the third and the last pile, he thinks of something different. He decided to collect only the minimum number of pens in the third pile so that the sum of pens in the three piles will give him a prime number.

Sample Input:

1 3

Sample Output:

1

```
1  #include<stdio.h>
2  #define MAX 1000
3  int is_prime(int n);
4  int main(){
5      int A, B;
6      scanf("%d%d", &A, &B);
7      int sum = A + B;
8      int i;
9      for(i = 1; i<=MAX; i++){
10         sum = sum + 1;
11         if(is_prime(sum)){
12             break;
13         }
14     }
15     printf("%d", i);
16 }
17
18
19
20
21
22
```

```
1  int is_prime(int n){
2      int is_prime = 1;
3      for(int i = 2; i <= n/2; i++){
4          if(n % i == 0){
5              is_prime = 0;
6              break;
7          }
8      }
9      return is_prime;
10 }
11
12
13
14
15
16
17
18
19
20
21
22
```

OUTPUT

```
778 916

3
```

Question 14

A man his driving car from home to the office with X petrol. There is an N number of petrol bunks in the city with only a few capacities and each petrol is located in different places. For one km one litre will consume. So he fills up petrol in his petrol tank in each petrol bunks. Output the remaining petrol if he has or tells him that he cannot travel if he is out of petrol. Write a program for the above-mentioned scenario.

Sample Input:

2
3
a b c
1 5 3
6 4 2

Sample Output:

5

```
#include<stdio.h>
int main(){
    int petrol_capacity, n;
    scanf("%d", &petrol_capacity);
    scanf("%d\n", &n);
    char petrol_bunk[n];
    for(int i = 0; i < n; i++){
        scanf("%c\n",&petrol_bunk[i]);
    }
    int  distance[n], capacities[n];
    for(int i = 0; i < n; i++){
        scanf("%d", &distance[i]);
    }
    for(int i = 0; i < n; i++){
        scanf("%d", &capacities[i]);
    }
```

```
1       for(int i = 0; i < n; i++){
2           if(petrol_capacity > 0){
3               petrol_capacity = (petrol_capacity - distance[i]) +
4                                                   capacities[i];
5           }
6           else{
7               break;
8           }
9       }
10    if(petrol_capacity > 0){
11      printf("%d", petrol_capacity);
12    }
13    else {
14      printf("No Petrol");
15    }
16  }
17
18
19
20
21
22
```

OUTPUT

```
2
3
A B C
1 5 3
2 1 1

No petrol
```

THANK YOU

FACE