



Objective:

- To test/challenge your grip on problem solving by involving Sequence, Decision and repetition structure.

Devise the solution of the following challenges using Pseudo code

Challenge-1: Candy Quest

(3)

You are a person who is always fond of eating candies. Your class fellow gave you a candy of length N , to eat during the break period of your school. You start eating this candy from one of the ends. But as it is not your candy, your friend told you to eat exactly K unit length of candy during each bite. You will stop eating if the candy's length becomes 0. It means that you have eaten the entire candy. If at some point of time, the candy's length is positive, but less than K , you *cannot* take a bite thereafter. Can you eat the complete candy? If yes, print the number of bites it will take, otherwise print "It is not possible to eat complete candy.".

Sample Run(s):

1	2
Enter Candy Length: 3 Enter Bite Length: 1 Minimum Bites Needed: 3	Enter Candy Length: 3 Enter Bite Length: 2 It is not possible to eat complete candy.

Challenge-2: Even Digit Sum

(3)

Write pseudo code to input a number and prints the sum of even digits in the number.

Sample Run(s):

1	2
Please Enter a Number: 3492 Sum of Even Digits is: 6	Please Enter a Number: 379 Sum of Even Digits is: 0

Challenge-3: Hockey Game Status

(4)

It's the hockey match finals in National Stadium Karachi and as always it has reached the penalty shotouts. Each team is given 5 shots to take and the team scoring a goal on the maximum number of shots wins the game. If both the teams' scores are equal, then the game is considered a draw and we would have 2 champions.

Write pseudo code which take ten values from the user (input value will be either 0 or 1, where 1 represents a goal and 0 represents a miss). First input value will represent the status of penalty shot by 1st team and second input value represents the penalty shot status by 2nd team, third input value represents the shot status of 1st team and so on.

After taking ten values from user, your pseudo should report the winning team or match draw status if none is won.

Sample Run(s):

1	2
Enter penalty shot 1 status: 1 Enter penalty shot 2 status: 1 Enter penalty shot 3 status: 1 Enter penalty shot 4 status: 1 Enter penalty shot 5 status: 0 Enter penalty shot 6 status: 1 Enter penalty shot 7 status: 0 Enter penalty shot 8 status: 1 Enter penalty shot 9 status: 1 Enter penalty shot 10 status: 1 Game is won by second team.	Enter penalty shot 1 status: 0 Enter penalty shot 2 status: 0 Enter penalty shot 3 status: 0 Enter penalty shot 4 status: 0 Enter penalty shot 5 status: 0 Enter penalty shot 6 status: 0 Enter penalty shot 7 status: 0 Enter penalty shot 8 status: 0 Enter penalty shot 9 status: 0 Enter penalty shot 10 status: 0 Game is drawn.

Challenge-4: Jerry, The Gambler!

(4)

Jerry has X dollars in debt to his bookie but currently has only Y dollars to his name. According to an astrologer, it's jerry's lucky day today and he will win every game he steps in. A golden chance for jerry to pay his debt ☺. Write a pseudo code which determines the minimum number of times you would have to go *double-or-nothing* at the roulette table (a gambling game) to pay your debt.

Assumption: Jerry go "all-in" on every bet and win every bet, and the casino allows bets of any value.



Sample Run(s):

1	2
Enter Amount that Jerry has to its Name: 100 Enter Jerry's Debt Amount: 1000 After 1 bet(s), jerry balance: 200 After 2 bet(s), jerry balance: 400 After 3 bet(s), jerry balance: 800 After 4 bet(s), jerry balance: 1600 Number of Bet(s): 4	Enter Amount that Jerry has to its Name: 700 Enter Jerry's Debt Amount: 500 Number of Bet(s): 0

balance : ~~100~~ 200 400 ~~800~~ 1600
debt : 1000
numOfBets : ~~0~~ 1 2 3 4
count : 1 2 3 4 5





Objective:

- To test/challenge your grip on problem solving skills using repetition structure.

Devise the solution of the following challenges using Pseudo code

Challenge-1: Marble Game

(5)

Write a pseudo code that allows two players to play the marble game. In the marble game, you ask the first player how many marbles to start with. Then, the game begins. The first player must take 1, 2 or 3 marbles. Then the second player goes and must take 1, 2 or 3 marbles. The winner is the player who takes the last marble. Allow two users to play this game and print out the winner (player #1 or player #2). Assume that both players enter valid inputs (1, 2 or 3, and they never try to take more marbles than there are in the pile.)

Sample Run

```
How many marbles will you be playing with? 10
Player # 1, there are 10 marbles left.
How many marbles will you take? 3
Player # 2, there are 7 marbles left.
How many marbles will you take? 2
Player # 1, there are 5 marbles left.
How many marbles will you take? 1
Player # 2, there are 4 marbles left.
How many marbles will you take? 1
Player # 1, there are 3 marbles left.
How many marbles will you take? 3
Player #1, you took the last marble and have won!
```

F22

total chocolate =
rupees/cost = chocolate
wrapper = chocolate (6)

Challenge-2: Chocolate Factory

Consider the following scenario:

Charlie once visited chocolate factory to buy some chocolates. Chocolate cost Rs. 3 each and he had only Rs. 45. He was disappointed. He wanted more. But when he reached the factory he was amazed. There was a scheme on that special day. If you return 3 wrappers of the chocolate you will get one chocolate for free. So how many chocolates can he buy?

Solution:

Charlie has Rs. 45 and each chocolate cost Rs. 3. So he buys only 15. But there is a scheme. He will return 15 wrappers and get 5 chocolates free. Then he will return 3 out of 5 and get one free. And he will again use one wrapper with the remaining 2 to get one more. So $15 + 5 + 1 + 1 = 22$.

Your Task:

Write a program, which ask user about the amount of rupees he has and cost per chocolate. In return your program will output the maximum amount of chocolates that can be bought considering the special offer discussed in above scenario.

Sample Run(s)

1	2
Enter Amount of Rupees: 45	Enter Amount of Rupees: 100
Enter the cost per Chocolate: 3	Enter the cost per Chocolate: 13
Maximum Chocolates that can be bought are: 22	Maximum Chocolates that can be bought are: 10

$17/3 = 5$

Challenge-3: Mirroring Number

(7)

The number 89 is the first integer with more than one digit that fulfills a special property which is as follows:

We get the same number if expressed in the form: $= 8^1 + 9^2 = 89$

The next number having this property is 135, See this property again: $135 = 1^1 + 3^2 + 5^3$

Write pseudo code which test whether a given number has such property or not.

Sample Run(s)

1	2
Enter Number: 175	Enter Number: 123
It's a Mirroring Number	It's not a Mirroring Number

Challenge-4: Bit Status

Your job is to list all of the numbers up to $2^N - 1$ that contains a 1 on given bit number in their binary representation of it.



For Example: If $N=4$ and bit number = 2, then your pseudo code will print [2, 3, 6, 7, 10, 11, 14, 15].
The binary numbers from 1 to 16 are:

4 3 2 1 (bit #)
0 0 0 1
0 0 1 0
0 0 1 1
0 1 0 0
0 1 0 1
0 1 1 0
0 1 1 1
1 0 0 0
1 0 0 1
1 0 1 0
1 0 1 1
1 1 0 0
1 1 0 1
1 1 1 0
1 1 1 1
 $2^3 2^2 2^1 2^0$

Sample Run(s)	
1	2
Enter Value of N: 4 Enter the Bit # : 2 2, 3, 6, 7, 10, 11, 14, 15,	Enter Value of N: 3 Enter the Bit # : 1 1, 3, 5, 7

1, 3, 5, 7
2, 3, 6, 7, 10, 11, 14, 15, 12
 $4 \sqrt{50}$
 -48
2

$\sqrt{45}$

4

2

$2^{(3)} - 1$

ans = 17
Repeat Until (ans $\leq 2^N$)

$2 \sqrt{1}$
 -1
1

Repeat Until (n \leq bit number)

ans $\div 2$

ans $\div 2$
ans $\div 2$

bit. must be

supers : 50 45
cost : 4 3

chocolate : ~~12~~ 15 18 21

total : ~~12~~ 16 17 18 20 21

wrappers : ~~12~~ 4 1 2 15 18 21 3

$\sqrt{3}$
 $2 \sqrt{3}$
 $1-1$
50

$2 \sqrt{15}$
 $2 \sqrt{7-1}$
 $2 \sqrt{3-1}$
 $1-1$

$12 + 4 + 1$

$2 \sqrt{2-0}$

$2 \sqrt{31}$
 $2 \sqrt{15-1}$
 $2 \sqrt{7-1}$
 $2 \sqrt{3-1}$
 $1-1$



Objective:

- Distributing the functionality keeping in view the atomicity, Presentation Layer and Business Layer and using by reference mechanism wherever needed.

Challenge - 1: Digital Watch

(12.5(0.5, 4, 2, 3, 3))

In this challenge, you will develop a series of functions which demonstrate some basic operations of a digital watch.

1	void printTimeIn24HoursFormat (int hours, int minutes, int seconds); Prints the received time on console in 24-hour format.
2	void printTimeIn12HoursFormat (int hours, int minutes, int seconds); Prints the received time on console in 12-hour format.
3	void incrementHours (int * hours, int * minutes, int * seconds, int increment = 1); Increment the value of hours as provided in increment parameter.
4	void incrementMinutes (int * hours, int * minutes, int * seconds, int increment = 1); Increment the value of minutes as provided in increment parameter.
5	void incrementSeconds (int * hours, int * minutes, int * seconds, int increment = 1); Increment the value of seconds as provided in increment parameter.

Note: To get full credit, develop a smart logic (make the machine do less work) to achieve the tasks listed in this challenge and closely observe and follow the sample run. Functions related to time increment does nothing if *incremental value* ≤ 0.

Sample Run	
Sample Code	Console Output
int hr1 = 4, min1 = 5, sec1 = 50; printTimeIn24HoursFormat (hr1, min1, sec1);	04:05:50
printTimeIn12HoursFormat (hr1, min1, sec1);	04:05:50 AM
int hr2 = 15, min2 = 2, sec2 = 45; printTimeIn12HoursFormat (hr2, min2, sec2);	03:02:45 PM
incrementHours (&hr2, &min2, &sec2, 4); printTimeIn24HoursFormat (hr2, min2, sec2);	19:02:45 PM
incrementMinutes (&hr1, &min1, &sec1, 100); printTimeIn12HoursFormat (hr1, min1, sec1);	05:45:50 AM

Challenge - 2: Change k^{th} Digit

(5)

Write a C++ program which inputs three integers n , k , and d , where n is possibly a negative *int*, k is a non-negative *int*, and d is a non-negative single digit (between 0 and 9 inclusive), and it changes the contents of identifier n such that the k^{th} digit is replaced with d . Counting starts at 0 and goes from right to left. It prints the updated value of n at the end. If value of k is invalid (think yourself that what is invalid for k) then function will simply return without doing anything.

Examples

1	If $n = 468$, $k = 0$, $d = 1$ then the number becomes 461.
2	If $n = 468$, $k = 1$, $d = 1$ then the number becomes 418.

The prototype of the function that will achieve this task is as follows:

```
void changeKthDigit(int * number, int kthDigit, int newDigit);
```

In order to achieve atomicity for the function `changeKthDigit`, following helper/utility function also needs to be complete.

```
int getNumberLength (int);  
int getLargestNDigitNumber (int);
```

You have to decide yourself that how the provided utility functions will be helpful for the `changeKthDigit`.

Sample Run	
Sample Code	Console Output
int a = 123456; changeKthDigit(&a, 1, 9); cout << a << '\n';	123496
cout << getNumberLength(123456) << " : " << getLargestNDigitNumber(6);	6 : 999999

**"Any Fool can write code that a computer can understand.
Good programmers write code that humans can understand."**

[... MARTIN FOWLER ...]



Objective:

- we shall be looking at problems related to one-dimensional Array structure manipulation.

To get full credit in each of the challenges, your solution should be atomic, should also follow rest of conventions as discussed so far in lectures and labs.

Challenge - 1: Missing Integer

(4)

Write a function which receives an array of type int and returns the smallest missing positive integral value in it.

Examples

1	If array = {1, 2, 0}	3 is returned.
2	If array = {3, 4, -1, 1}	2 is returned.
3	If array = {7, 8, 9, 11, 12}	1 is returned.

Challenge - 2: Friend Request

(6)

You have been recently hired by Facebook and your first task is to analyze possible impact of its new expected policy regarding sending friend requests.

The new expected policy to be employed is as follows.

A person a cannot send a friend request to person b if any of the following condition is true:

- age of $b \leq (0.5 * \text{age of } a + 7)$
- age of $b > \text{age of } a$
- (age of $b > 100$) and (age of $a < 100$)

Facebook will provide you an array of integers whose each location represents the age of their user. So, if the array name is ages, then ages[i] represents the age of user i where $0 \leq i \leq N$.

Note:

- A person can't send request to itself.
- If a sends request to b then it is not necessary for b to send request to a .
- A person age will always be in the range $1 \leq \text{age} \leq 120$.

So, your job is to find the total number of friend requests possible under the new policy. Your function results will help upper management to decide something about implementing the new policy. For this you are given following function which receives ages array and its size and returns the required information.

```
int getFriendRequestsCount (int ages [], int N);
```

Examples

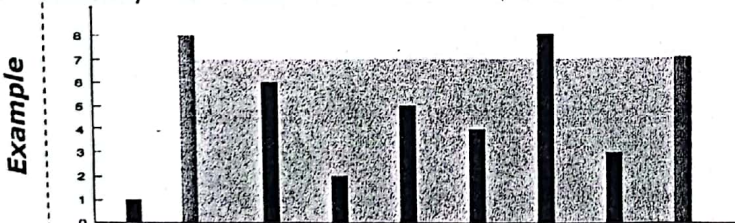
1	If ages array = {20, 20}	2 is returned.
2	If ages array = {20, 30, 40}	3 is returned.

Challenge - 3: Container Capacity

(10)

Given N non-negative integers a_1, a_2, \dots, a_n , where each represents a point at coordinate (i, a_i) . Total N vertical lines are drawn such that the two endpoints of line i is at $(i, 0)$ and (i, a_i) . Find two lines, which together with x -axis forms a container, such that the container contains the most water.

Note: You may not slant the container and N is at least 2.



The vertical lines are represented by array {1, 8, 6, 2, 5, 4, 8, 3, 7}. In this case, the max area of water (grey shaded section) the container can contain is 49.

To achieve above task, you need to implement following function.

```
int getMaxWaterArea(int arr[], int N);
```

Magic happens when you don't give up, even though you want to. The universe always falls in love with a stubborn heart.

[... ...]



Objective:

- Looking at an application of 1D Arrays: IEEE-754 Single Precision Conversion.

IEEE-754 Conversion

Write a series of function as listed below: Their main goal is to implements the IEEE-754 single precision procedure as discussed in class.

1. float getFloatingPointValueIEEE754(bool []); (6)
 - returns the floating-point value represented by received array.
 - Received array size is 32.
 - Index 0 of array represents the bit number 31 i.e., most significant bit (msb) and index 31 represents the bit number 0 i.e., least significant bit (lsb).
2. void convertFloatingPointValueTo32BitRepresentationIEEE754(float, bool []); (12)
 - populate the received array with 32-bit representation of received floating point value as per IEEE-754.
3. void print32BitRepresentationIEEE754(bool []); (2)
 - prints the received array.

Note:

- Don't use any short cut to achieve the above tasks.
- You are required to implement procedure as discussed in class.
- You may define additional functions in order to make your work atomic and manageable.
- To get full marks, your console output should be similar to the sample run.

Sample Run

```
1 int main()
2 {
3     bool arr[32] =
4         {0, 1,0,0,0, 0,0,1,0, 1,0,0, 0,1,1,0, 0,0,0,0, 0,0,0,0, 0,0,0,0, 0,0,0,0};
5     print32BitRepresentationIEEE754(arr);
6     float val = getFloatingPointValueIEEE754(arr);
7     cout << "\nFloating Point Value = " << val;
8     bool res[32];
9     convertFloatingPointValueTo32BitRepresentationIEEE754 (12.375, res);
10    print32BitRepresentationIEEE754(res);
11    return 0;
12 }
```

Console Output

```
4 SB Exponent Mantissa
0 1000 0010 100 0110 0000 0000 0000 0000
6 Floating Point Value = 12.375
9 SB Exponent Mantissa
0 1000 0010 100 0110 0000 0000 0000 0000
```

Just a funny story about Steve Jobs and Floating Point Numbers

"Steve Jobs calls one of the mysteries of life how his friend and Apple's Co-founder Steve Wozniak never built a floating point BASIC for the Apple II, despite Jobs having begged him for several weeks to do it. As a result of Wozniak's refusal, Job reached out to Microsoft to license Bill Gates' BASIC."

https://en.wikipedia.org/wiki/Applesoft_BASIC

"There are no traffic jams along the extra mile."

[... ROGER STAUBACH ...]



Objective:

- Looking at problems related to 2-dimensional arrays.

Challenge - 1: Most Loveable Food

(5)

A restaurant wants to know their most loved food/menu-item in their menu. For this, they surveyed from N customers about their M dishes in the menu. Your job is to analyze the survey and compute the most loveable food mentioned in it.

Restaurant will provide you a matrix of N by M where N represents the number of customers and M represents the dishes. If a dish is liked a customer, then the matrix entry will be true otherwise false.

Example: In the sample matrix below: there are 5 customers and 8 dishes and you can see that dish number 2 is most loveable dish.

	0	1	2	3	4	5	6	7
0	true	false	true	false	true	false	false	false
1	false	false	true	true	false	false	true	false
2	false	false	false	true	true	false	false	false
3	true	true	true	true	true	true	true	true
4	false	false	true	false	false	false	false	true

Function Prototype: int getMostLoveableDish(bool mat[][], int N, int M);

Challenge - 2: Rotate Matrix

(8)

Given an N order matrix, clockwise rotate elements in it.

Function Prototype: void rotateMatrix(int mat[][], int N);

Examples/Explanation	
For 3*3 matrix 1 2 3 4 5 6 7 8 9	Matrix Elements After Rotation: 4 1 2 7 5 3 8 9 6
For 4*4 matrix 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16	Matrix Elements After Rotation: 5 1 2 3 9 10 6 4 13 11 7 8 14 15 16 12

سوداگری نہیں یہ عبادت خدا کی ہے
اے بے خبر جزا کی تمنا بھی چھوڑ دے
واعظ ثبوت لائے جو مے کے جواز میں
اقبال کو یہ ضد ہے کہ پینا بھی چھوڑ دے

[ڈاکٹر علامہ محمد اقبال]