Name:- Anshumali Karna Roll No.:- 20103291 Batch:- B9

Q1.

**import** **java.util.\***;

**class** Node {

**int** value;

**Node** left;

**Node** right;

Node(**int** value) {

**this**.value **=** value;

right **=** **null**;

left **=** **null**;

}

}

**public** **class** QuestionOne {

**public** **static** **int** minRounds(**Node** root) {

**if** (root **==** **null**) {

**return** 0;

}

**int** left **=** minRounds(root.left);

**int** right **=** minRounds(root.right);

**return** Math.max(left, right) **+** 1;

}

**public** **static** **void** main(**String**[] args) {

**Node** root **=** **new** Node(0);

root.left **=** **new** Node(1);

root.right **=** **new** Node(2);

root.left.left **=** **new** Node(3);

root.left.right **=** **new** Node(4);

root.right.left **=** **new** Node(5);

root.right.right **=** **new** Node(6);

root.left.left.left **=** **new** Node(7);

root.left.left.right **=** **new** Node(8);

root.left.right.left **=** **new** Node(9);

root.left.right.right **=** **new** Node(10);

root.right.left.left **=** **new** Node(11);

root.right.left.right **=** **new** Node(12);

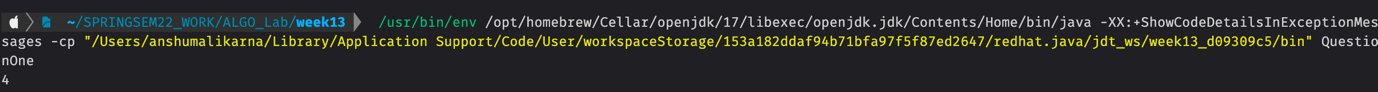
root.right.right.left **=** **new** Node(13);

root.right.right.right **=** **new** Node(14);

System.out.println(minRounds(root));

}

}

OUTPUT:-

Q2.

**public** **class** QuestionTwo {

**public** **static** **int** maxValue(**String** s) {

**int** max **=** 0;

**int** curr **=** 0;

**int** sign **=** 1;

**for** (**int** i **=** 0; i **<** s.length(); i**++**) {

**if** (s.charAt(i) **==** '+') {

sign **=** 1;

} **else** **if** (s.charAt(i) **==** '-') {

sign **=** **-**1;

} **else** {

curr **=** curr **\*** 10 **+** (s.charAt(i) **-** '0');

max **=** Math.max(max, curr **\*** sign);

}

}

**return** max;

}

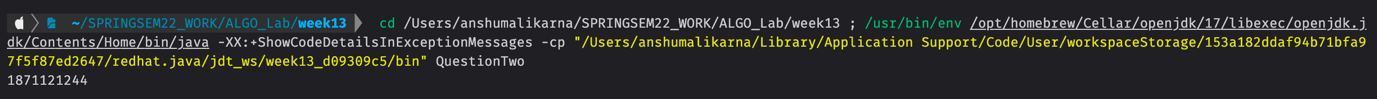
**public** **static** **void** main(**String**[] args) {

System.out.println(maxValue("1+ 3 − 2 − 5 + 1 − 6 + 7"));

}

}

OUTPUT:-



Q3.

**#include**<Foundation/Foundation.h>

**@interface** QuestionThree : NSObject

-(**int**)findLongestSubstringWithKDistinctVowels:(NSString **\***)s k:(**int**)k;

**@end**

**@implementation** QuestionThree

-(**int**)findLongestSubstringWithKDistinctVowels:(NSString **\***)s k:(**int**)k{

NSMutableDictionary **\***dict **=** [NSMutableDictionary dictionary];

**int** maxLength **=** 0;

**int** start **=** 0;

**int** end **=** 0;

**int** count **=** 0;

**while** (end **<** s.length) {

NSString **\***character **=** [s substringWithRange:NSMakeRange(end, 1)];

**if** ([character isEqualToString:@"a"] **||** [character isEqualToString:@"e"] **||** [character isEqualToString:@"i"] **||** [character isEqualToString:@"o"] **||** [character isEqualToString:@"u"]) {

count**++**;

}

end**++**;

**if** (count **==** k) {

maxLength **=** MAX(maxLength, end **-** start);

}

**if** (count **>** k) {

NSString **\***startCharacter **=** [s substringWithRange:NSMakeRange(start, 1)];

**if** ([startCharacter isEqualToString:@"a"] **||** [startCharacter isEqualToString:@"e"] **||** [startCharacter isEqualToString:@"i"] **||** [startCharacter isEqualToString:@"o"] **||** [startCharacter isEqualToString:@"u"]) {

count**--**;

}

start**++**;

}

}

**return** maxLength;

}

**@end**

**int** main() {

@autoreleasepool {

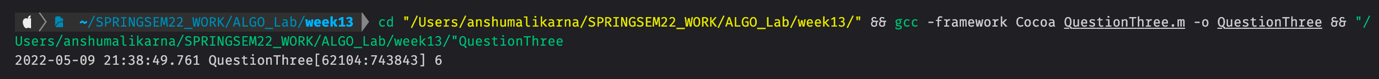
QuestionThree **\***obj **=** [[QuestionThree alloc] init];

NSLog(@"%d", [obj findLongestSubstringWithKDistinctVowels:@"artyebui" k:2]);

}

**return** 0;

}



Q4.

**#include**<Foundation/Foundation.h>

**@interface** QuestionFour : NSObject

-(**int**)findContiguousSubsequenceWithMaximumSum:(NSArray **\***)array;

**@end**

**@implementation** QuestionFour

-(**int**)findContiguousSubsequenceWithMaximumSum:(NSArray **\***)array{

**int** maxSum **=** 0;

**int** currentSum **=** 0;

**for** (**int** i **=** 0; i **<** array.count; i**++**) {

currentSum **+=** [array[i] intValue];

**if** (currentSum **<** 0) {

currentSum **=** 0;

}

**if** (currentSum **>** maxSum) {

maxSum **=** currentSum;

}

}

**return** maxSum;

}

**@end**

**int** main() {

@autoreleasepool {

QuestionFour **\***obj **=** [[QuestionFour alloc] init];

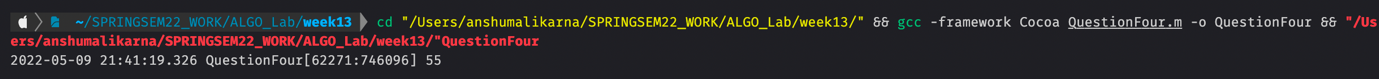
NSLog(@"%d", [obj findContiguousSubsequenceWithMaximumSum:@[@5, @15, @**-**30, @10, @**-**5, @40, @10]]);

}

**return** 0;

}

OUTPUT:-



Q5.

**#include**<Foundation/Foundation.h>

**@interface** QuestionTwo : NSObject

-(**int**)findMinimumEdits:(NSString **\***)src dest:(NSString **\***)dest;

**@end**

**@implementation** QuestionTwo

-(**int**)findMinimumEdits:(NSString **\***)src dest:(NSString **\***)dest{

**int** srcLength **=** (**int**)src.length;

**int** destLength **=** (**int**)dest.length;

**int** **\*\***dp **=** (**int** **\*\***)malloc(**sizeof**(**int** **\***) **\*** (srcLength **+** 1));

**for** (**int** i **=** 0; i **<=** srcLength; i**++**) {

dp[i] **=** (**int** **\***)malloc(**sizeof**(**int**) **\*** (destLength **+** 1));

**for** (**int** j **=** 0; j **<=** destLength; j**++**) {

dp[i][j] **=** 0;

}

}

**for** (**int** i **=** 1; i **<=** srcLength; i**++**) {

dp[i][0] **=** i;

}

**for** (**int** i **=** 1; i **<=** destLength; i**++**) {

dp[0][i] **=** i;

}

**for** (**int** i **=** 1; i **<=** srcLength; i**++**) {

**for** (**int** j **=** 1; j **<=** destLength; j**++**) {

**if** ([src characterAtIndex:i **-** 1] **==** [dest characterAtIndex:j **-** 1]) {

dp[i][j] **=** dp[i **-** 1][j **-** 1];

}**else**{

dp[i][j] **=** MIN(dp[i **-** 1][j], dp[i][j **-** 1]) **+** 1;

}

}

}

**return** dp[srcLength][destLength];

}

**@end**

**int** main() {

@autoreleasepool {

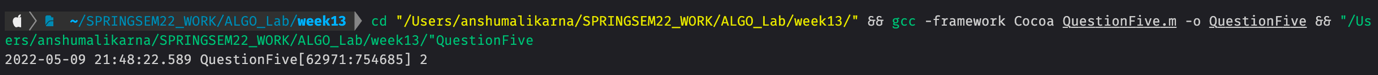
QuestionTwo **\***obj **=** [[QuestionTwo alloc] init];

NSLog(@"%d", [obj findMinimumEdits:@"abcd" dest:@"abce"]);

}

**return** 0;

}



Q6.

**#include**<Foundation/Foundation.h>

**@interface** QuestionSix : NSObject

-(**void**)MatrixChainOrder:(NSArray **\***)array;

**@end**

**@implementation** QuestionSix

-(**void**)MatrixChainOrder:(NSArray **\***)array{

NSMutableArray **\***dp **=** [NSMutableArray array];

**for** (**int** i **=** 0; i **<** array.count; i**++**) {

[dp addObject:@0];

}

**for** (**int** i **=** 1; i **<** array.count; i**++**) {

**for** (**int** j **=** 0; j **<** i; j**++**) {

**int** min **=** INT\_MAX;

**for** (**int** k **=** j; k **<** i; k**++**) {

min **=** MIN(min, [dp[k] intValue] **+** [array[j] intValue] **\*** [array[k **+** 1] intValue]);

}

dp[i] **=** [NSNumber numberWithInt:min];

}

}

NSLog(@"%d", [dp[array.count **-** 1] intValue]);

}

**@end**

**int** main() {

@autoreleasepool {

QuestionSix **\***obj **=** [[QuestionSix alloc] init];

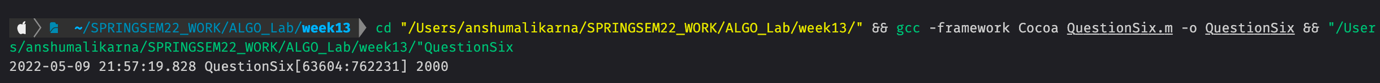
[obj MatrixChainOrder:@[@40, @20, @30, @10, @30]];

}

**return** 0;

}

OUTPUT:-



Q7.

**#include** <Foundation/Foundation.h>

**@interface** QuestionSeven : NSObject

-(NSArray **\***)findLongestCommonSubsequence:(NSArray **\***)s1;

**@end**

**@implementation** QuestionSeven

-(NSArray **\***)findLongestCommonSubsequence:(NSArray **\***)s1{

NSMutableArray **\***dp **=** [NSMutableArray array];

**for** (**int** i **=** 0; i **<** s1.count; i**++**) {

[dp addObject:@0];

}

**for** (**int** i **=** 1; i **<** s1.count; i**++**) {

**for** (**int** j **=** 0; j **<** i; j**++**) {

**if** ([s1[i] isEqual:s1[j]]) {

dp[i] **=** [NSNumber numberWithInt:[dp[i] intValue] **+** 1];

}

}

}

**int** max **=** 0;

**for** (**int** i **=** 0; i **<** dp.count; i**++**) {

**if** ([dp[i] intValue] **>** max) {

max **=** [dp[i] intValue];

}

}

NSMutableArray **\***result **=** [NSMutableArray array];

**for** (**int** i **=** 0; i **<** dp.count; i**++**) {

**if** ([dp[i] intValue] **==** max) {

[result addObject:s1[i]];

}

}

**return** result;

}

**@end**

**int** main() {

@autoreleasepool {

QuestionSeven **\***obj **=** [[QuestionSeven alloc] init];

NSLog(@"%@", [obj findLongestCommonSubsequence:@[@"B", @"C", @"D", @"A", @"A", @"C", @"D"]]);

}

**return** 0;

}

OUTPUT:-

