CSC207 · INTRODUCTION TO ALGORITHMS · CA 2025

CSC207
CA 2025

Every question on the CSC207 CA 2025 paper, word for word, solved in full in the course pseudocode: ordered set snippets, isPalindrome, exact-output tracing, and the course registration pseudocode with its flowchart.

Section A and B | 6 worked solutions | Every output checked

Solved by **Tabe Rickson** · taberickson.com/csc207/ca2025
Based on the CSC207 Introduction to Algorithms CA 2025 paper.

What is inside

SECTION A · ANSWER ALL QUESTIONS

A1 initSet() and contains()

A2 isPalindrome()

A3a Exact output: switch

A3b Exact output: for loop

A3c Exact output: while with %

SECTION B · PSEUDOCODE AND FLOWCHART

B1 Course registration and ranking

HOW TO USE THIS DOCUMENT

The **blue box** is the question exactly as it is on the paper. The solution follows it directly.

`read(x)` reads a value, `println(...)` prints and ends the line, `print(...)` stays on the same line.

SECTION A

Answer all questions

A1 initSet() and contains()

QUESTION A1

1. A set is a data structure which helps us to collect items/objects of the same type/domain. An ordered set is one whose position of its elements is relevant. Consider the following structure for an ordered set:

```
struct OrderedSet{
	int data[MAX_SIZE]; // store elements
	int size;

};
```

Propose the code snippet(s) ONLY that is at position (a) and (b) for the methods to initialize an ordered set (initSet()) or check if an ordered set contains some element (contains()). Initializing a set means deleting/clearing all its elements/data, while "contains" is to check if a value is an element of the set. (2+3 = 5 marks)

```
void initSet(OrderedSet *set) {
	___(a)___

}

bool contains(OrderedSet *set, int value) {
	for (int i = 0; i < set->size; i++) {
		___(b)___
	}
	return false;

}
```

SOLUTION

(a)

```
set->size = 0;    // no elements left: the set is cleared
```

(b)

```
if (set->data[i] == value)
    return true;    // found: stop and answer yes
```

In place:

```
void initSet(OrderedSet *set) {
    set->size = 0;                    // (a)
}
```

```c
bool contains(OrderedSet *set, int value) {
    for (int i = 0; i < set->size; i++) {
        if (set->data[i] == value)    // (b)
            return true;
    }
    return false;
}
```

A2 isPalindrome()

QUESTION A2

2. Consider the following structure for a word:

```
struct Word{
       char elements[MAX];
       int length;

};
```

A word is a palindrome if it is exactly the same as the word itself, when written in the reversed order. For example: “dad” and “mom” are palindromes. “grace” is not a palindrome because the reversed word order gives “ecarg”. Assume the characters are in lower case. Propose an algorithm called isPalindrome() with appropriate input parameter(s) and returned value, which can be used to determine if a word is a palindrome or not. (4 marks)

SOLUTION

```c
bool isPalindrome(Word *w) {
    int left = 0;
    int right = w->length - 1;
    while (left < right) {
        if (w->elements[left] != w->elements[right])
            return false;             // a mismatch: not a palindrome
        left++;
        right--;
    }
    return true;                      // every pair matched
}
```

Input: the word (a pointer to `Word`). Returns `true` if it is a palindrome, `false` otherwise. “dad” → true, “grace” → false (g ≠ e).

A3a Exact output: switch

QUESTION A3A

3. Carefully study the algorithms below and write down the “exact” displayed output. Put your final output for each sub-question in a box/frame. (2 × 3 = 6 marks)

(a)

```
today= 5;
switch(today)
{
        case 1: print("stay home");break;
        case 2: print("programing");break;
        case 3: print("mathematics");break;
        default: print(today);
}
```

SOLUTION

`today` is 5: no case matches, so `default` prints the value of `today`.

EXACT OUTPUT

```
5
```

A3b Exact output: for loop

QUESTION A3B

(b)

```
i = 1;
for(j=1; j<10; j=j+2){
        println(i);
}
println(j);
```

SOLUTION

`j` takes 1, 3, 5, 7, 9: five rounds, each prints `i`, which stays 1. The loop stops when `j` = 11, and the last line prints it.

EXACT OUTPUT

```
1
1
1
1
1
11
```

A3c Exact output: while with %

QUESTION A3C

(c)

```
n=10;
i=1;
while(i<=n){
        if(i%3 == 0){
                println(i);
        }
        i++;
}
```

SOLUTION

`i` runs from 1 to 10, and only the multiples of 3 are printed.

EXACT OUTPUT

```
3
6
9
```

SECTION B

Pseudocode and flowchart

B1 Course registration and ranking

QUESTION B1

Section B (Answer on a separate sheet)

1. Consider the following scenario: A student is asked for the students' name and how many courses registered. If the student registered up to 4 courses, grade the student, and above four courses notify the student to drop excess courses and register only for four.

Ranking of students is done as follows: for average greater than 90 is "Excellent", average greater than 80 "Very Good", average greater than 60 "Good" and otherwise failed.

(a) Write the Pseudocode for the above problem. (b) Draw the corresponding Flow chart. (8+10 = 18 marks)

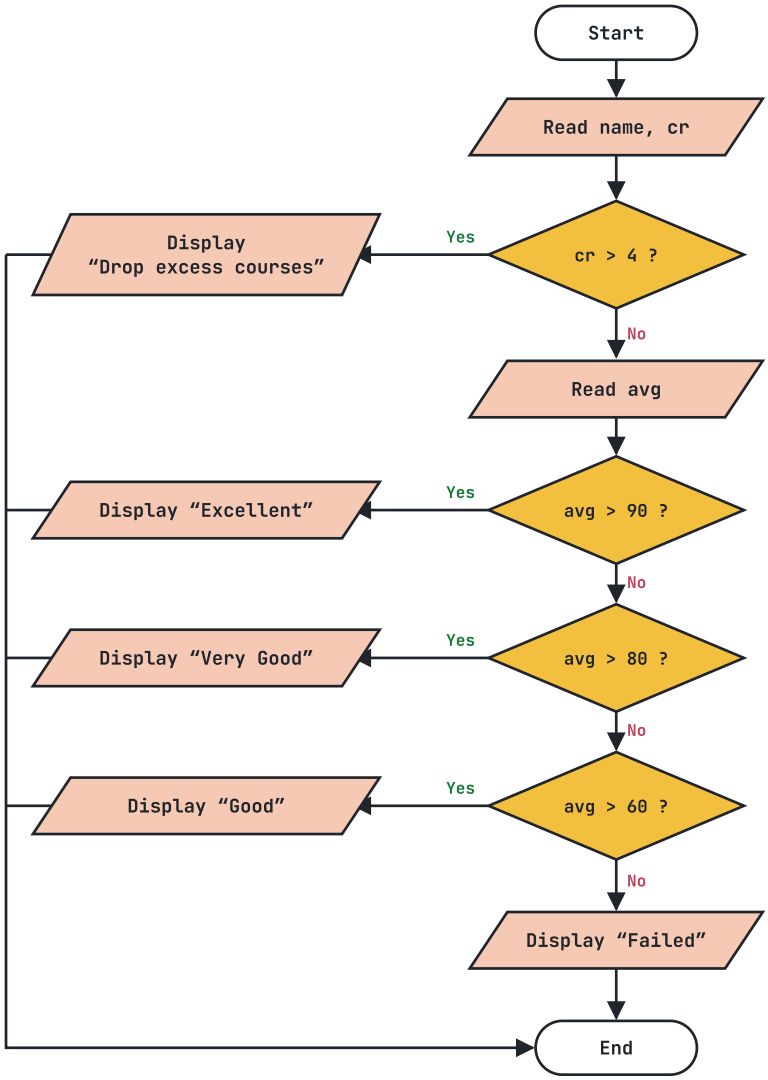
SOLUTION

(a) Pseudocode

```
void gradeStudent() {
    char name[50];
    int cr;                          // courses registered
    float avg;                       // the student's average

    read(name);
    read(cr);
    if (cr > 4) {
        println(name, ": drop excess courses and register only 4");
    } else {
        read(avg);
        if (avg > 90)
            println("Excellent");
        else if (avg > 80)
            println("Very Good");
        else if (avg > 60)
            println("Good");
        else
            println("Failed");
    }
}
```

(b) Flowchart



Flowchart for Section B, Question 1

Trace by hand, line by line, and write what is printed exactly as it would appear. That is most of the marks in Section A.