Princess Sumaya جامعة الأميرة سميّة للتكنولوجيا University for Technology

The King Hussein School for Computing Sciences
Department of Computer Science
Structured Programming - Fall 2021

Midterm Exam

Full Name: **Student ID:**

Question	Points	Score
1	10	
2	10	
3	15	
4	15	
5	20	
6	20	
Total	100	

Circle your section:

- ○ Dr. Ahmad AlNabhan (section 1)
- ○ Dr. Rawan Ghonaimat (section 2)
- ○ Dr. Rawan Ghonaimat (section 3)
- ○ Dr. Mohammad Abu Snober (section 4)
- ○ Dr. Abdullah Aref (section 5)
- ○ Dr. Sawsan AlShatnawi (section 6)
- ○ Dr. Ahmad AlNabhan (section 7)

Question 1 (10 points)

```
for (int i = 0; i < n; i++) {
    for (int j = 0; j < i; j++)
        a[i] = a[i] + 2;
}
```

Assuming that `a[]` is initialized to zeros, what is the value of `a[n-1]` after executing the code on the left in each of the following cases?

- If `n = 1`
- If `n = 2`
- If `n = 101`

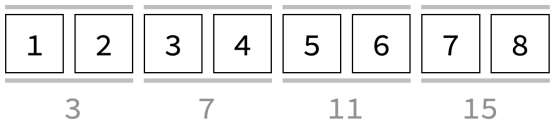
Solution.

The code stores at `a[i]` the value of `2*i`.

Question 2 (10 points)

Implement function `sum_pairs(int a[], int size)`, which prints the sum of pairs in the given array as shown in the illustration and sample output below. If the `size` of the array is not a multiple of 2, output an error message and exit the function.

Example.



Output.

```
1 + 2 = 3
3 + 4 = 7
5 + 6 = 11
7 + 8 = 15
```

Solution.

```
void sum_pairs(int a[], int size) {
    if (size % 2 != 0) {
        printf("Invalid array size\n");
        return;
    }
    for (int i = 0; i < size; i += 2)
        printf("%d + %d = %d\n", a[i], a[i+1], a[i]+a[i+1]);
}
```

Question 3 (15 points)

```
int num, size, temp1 = -1, temp2 = -1;
scanf("%d", &size);

for (int i = 0; i < size; i++) {
     scanf("%d", &num);
     if (i == 0)
          temp1 = num;
     else if (num > temp1) {
          temp2 = temp1;
          temp1 = num;
     } else if (num > temp2)
          temp2 = num;
}

printf("%d", temp2);
```

What does the above code print for each of the following input sequences?

- `4 1 2 1 1`
- `1 1`
- `1000 1 2 3 4 5 … 997 998 999 1000`
- `100 100 99 98 97 … 5 4 3 2 1`
- `1000` followed by the numbers from `1` to `1000` but in random order.

Solution.

> The code prints the second largest element in the input and prints `-1` if the sequence has `<= 1` elements.

Question 4 (15 points)

The instructors of the structured programming lab decided to consider for the final grade the top 11 labs only out of the 12 labs students did during the semester. Write a complete C program that reads the grades for the 12 labs and then prints the average grade for the highest 11 labs.

Solution.

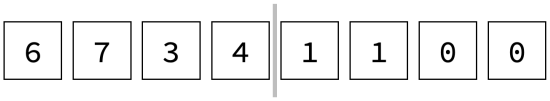
```c
#include <stdio.h>

int main() {
	int sum = 0;
	int min, grade;

	for (int i = 0; i < 12; i++) {
		scanf("%d", &grade);
		if (i == 0 || grade < min)
			min = grade;
		sum += grade;
	}

	printf("%f", (sum - min) / 11.0);
	return 0;
}
```

Question 5 (20 points)

Implement function `int check(int a[], int size)`, which returns `1` if any of the elements in the first half of the given array is also in the second half. The function returns `0` otherwise.

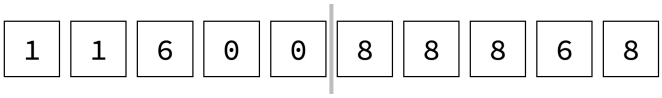
You can assume that the array size is even (no need to check for this).

Example 1.



The function returns `0`, because none of the numbers in the first half is present in the second half.

Example 2.



The function returns `1`, because `6` from the first half is present in the second half.

Solution.

```
int check(int a[], int size) {
      for (int i = 0; i < size / 2; i++) {
            for (int j = size / 2; j < size; j++)
                  if (a[i] == a[j])
                        return 1;
      }
      return 0;
}
```

Question 6 (20 points)

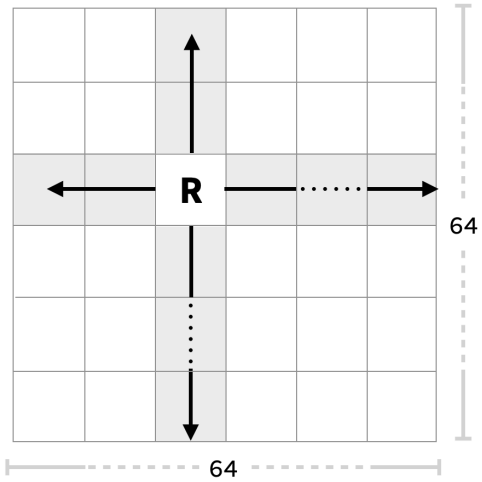
Implement function

`void check(char board[][64], int i, int j)`

which receives a chess board of size `64x64` and the position of a rook (♖ قلعة).

The function prints `'yes'` if there is another rook in its same column or in its same row and `'no'` otherwise.

Assume that the board contains the character `'R'` in a cell if there is a rook in that cell.



Solution.

```
void check(char board[][64], int i, int j) {
      // check same row
      for (int k = 0; k < 64; k++)
            if (board[i][k] == 'R' && k != j) {
                  printf("yes");
                  return;
            }

      // check same column
      for (int k = 0; k < 64; k++)
            if (board[k][j] == 'R' && k != i) {
                  printf("yes");
                  return;
            }

      printf("no");
}
```