CSE 340 Midterm 1

Olcay Taner YILDIZ

I. QUESTION (10 POINTS)

Write the method

```
int sum_until()
```

which is going to read a sequence of integers. You will know the end of the sequence when a number is equal to the sum of the two numbers before it. Your method will print the sum of integers in the sequence including the last number. You can assume that at least three numbers will be entered.

II. QUESTION (15 POINTS)

A sequence of $n > 0$ integers is called a jolly jumper if the absolute values of the differences between successive elements take on all possible values 1 through n - 1. For instance,

1 3 6 5

is a jolly jumper, because the absolute differences are 2, 3, and 1, respectively. The differences are all possible numbers 1 through 3.

5 2 1 5 3

is again a jolly jumper, because the absolute differences are 3, 1, 4, and 2 respectively. The differences are all possible numbers 1 through 4.

3 4 2 6

is not a jolly jumper, because the absolute differences are 1, 2, and 4 respectively. The differences are not all possible numbers 1 through 3. The definition implies that any sequence of a single integer is a jolly jumper. Write the method

```
int jolly_jumper(int* array, int size)
```

to determine whether a sequence is a jolly jumper. The function will return 1, if the sequence is jolly jumper, 0 otherwise.

III. QUESTION (15 POINTS)

Write the method

```
int* double_list(int* array, int N)
```

which doubles each number in an integer array of size N, that is, after each element inserts that element again. The result should be a new array of size $2N$.

IV. QUESTION (20 POINTS)

Write the method

```
int** dynamic_array(int rows, int* columns)
```

which allocates, fills and returns a two dimensional array, where row i has $columns[i]$ columns. Write also

```
void free_dynamic_array(int** array, int rows)
```

which deallocates memory allocated for that array.

V. QUESTION (10 POINTS)

Declare struct **club** which contains name (String), player_count (integer), player_names (constant String array of size 11), point (integer). Write also method

```
Clubptr new_club(char* name)
```

which creates and returns a new club pointer from the given parameters. player_count and point should be initialized to 0.

VI. QUESTION (10 POINTS)

Write the method

```
void play_match(Clubptr club1, Clubptr club2)
```

that simulates a match played between club1 and club2. The number of goals scored by the club1 and club2 are taken from the user. If club1 wins the match, club1s point is increased by 3, else if club2 wins the match, club2s point is increased by 3, else (in case of a tie) both club1 and club2s points are increased by 1.

VII. QUESTION (20 POINTS)

Write methods

```
void add_player(Clubptr club, char* name)
void remove_player(Clubptr club, char* name)
```

which adds/removes player to/from the club with the given player name.