

SE 322 Midterm #1

Olcay Taner YILDIZ

I. QUESTION (15 POINTS)

Write the method

```
void 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)

Write the method

```
void prime_factors(int N)
```

which prints the prime factors of N . Each prime factor should be printed only once.

III. QUESTION (15 POINTS)

Write a function to add two matrices. The first matrix will have the result in it. The function prototype is as follows:

```
int** addMatrix(int** first,
                int** second,
                int rows, int cols)
```

The parameters `rows` and `cols` show the number of rows and columns of the first and second matrices.

IV. QUESTION (20 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. You can assume that the size of the array is less than 100 (No need for dynamic allocation).

V. QUESTION (15 POINTS)

Write the function

```
void strncat(char* destination,
             char* source,
             int num)
```

which appends the first `num` characters of `source` to `destination`, plus a terminating null-character. If the length of the string in `source` is less than `num`, only the content up to the terminating null-character is copied.

VI. QUESTION (20 POINTS)

Write the function

```
int atoi(char *source)
```

which converts the string in `source` to an integer.