Міністерство освіти і науки України
Національний технічний університет України «КПІ ім. Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Мультипарадигменне програмування

ЗВІТ

до лабораторної роботи №2

Виконав студент ІП-01 Пасальський Олександр _____
(№ групи, прізвище, ім'я, по батькові)

Прийняв ас. Очеретяний О. К. _____
(посада, прізвище, ім'я, по батькові)

Київ 2022

1. Завдання лабораторної роботи

Завдання

Ви напишете 11 функцій SML (і тести для них), пов’язаних з календарними датами. У всіх завданнях, **“дата”** є значенням SML типу `int*int*int`, де перша частина - це рік, друга частина - місяць і третя частина - день. **«Правильна»** дата має позитивний рік, місяць від 1 до 12 і день не більше 31 (або 28, 30 - залежно від місяця). Перевіряти “правильність” дати не обов'язково, адже це досить складна задача, тож будьте готові до того, що багато ваших функцій будуть працювати корректно для деяких/всіх **“неправильних”** дат у тому числі. Також, **«День року»** — це число від 1 до 365 де, наприклад, 33 означає 2 лютого. (Ми ігноруємо високосні роки, за винятком однієї задачі.)

1. Напишіть функцію is_older, яка приймає дві дати та повертає значення true або false. Оцінюється як true, якщо перший аргумент - це дата, яка раніша за другий аргумент. (Якщо дві дати однакові, результат хибний.)

2. Напишіть функцію number_in_month, яка приймає список дат і місяць (тобто int) і повертає скільки дат у списку в даному місяці.

3. Напишіть функцію number_in_months, яка приймає список дат і список місяців (тобто список int) і повертає кількість дат у списку дат, які знаходяться в будь-якому з місяців у списку місяців. **Припустимо, що в списку місяців немає повторюваних номерів**. Підказка: скористайтеся відповіддю до попередньої задачі.

4. Напишіть функцію dates_in_month, яка приймає список дат і число місяця (тобто int) і повертає список, що містить дати з аргументу “список дат”, які знаходяться в переданому місяці. Повернутий список повинен містять дати в тому порядку, в якому вони були надані спочатку.

5. Напишіть функцію dates_in_months, яка приймає список дат і список місяців (тобто список int) і повертає список, що містить дати зі списку аргументів дат, які знаходяться в будь-якому з місяців у списку місяців. Для простоти, припустимо, що в списку місяців немає повторюваних номерів. Підказка: Використовуйте свою відповідь на попередню задачу та оператор додавання списку SML (@).

6. Напишіть функцію get_nth, яка приймає список рядків і int n та повертає n-й елемент списку, де голова списку є першим значенням. Не турбуйтеся якщо в списку занадто мало елементів: у цьому випадку ваша функція може навіть застосувати hd або tl до порожнього списку, і це нормально.

7. Напишіть функцію date_to_string, яка приймає дату і повертає рядок у вигляді “February 28, 2022” Використовуйте оператор ^ для конкатенації рядків і бібліотечну функцію Int.toString для перетворення int в рядок. Для створення частини з місяцем не використовуйте купу розгалужень. Замість цього використайте список із 12 рядків і свою відповідь на попередню задачу. Для консистенції пишіть кому після дня та використовуйте назви місяців англійською мовою з великої літери.

8. Напишіть функцію number_before_reaching_sum, яка приймає додатний int під назвою sum, та список int, усі числа якої також додатні. Функція повертає int. Ви повинні повернути значення int n таке, щоб перші n елементів списку в сумі будуть менші sum, але сума значень від n + 1 елемента списку до кінця був більше або рівний sum.

9. Напишіть функцію what_month, яка приймає день року (тобто int між 1 і 365) і повертає в якому місяці цей день (1 для січня, 2 для лютого тощо). Використовуйте список, що містить 12 цілих чисел і вашу відповідь на попередню задачу.

10. Напишіть функцію month_range, яка приймає два дні року day1 і day2 і повертає список int [m1,m2,...,mn] де m1 – місяць day1, m2 – місяць day1+1, ..., a mn – місяць day2. Зверніть увагу, що результат матиме довжину day2 - day1 + 1 або довжину 0, якщо day1>day2.

11. Напишіть `найстарішу` функцію, яка бере список дат і оцінює параметр (int*int*int). Він має оцінюватися як NONE, якщо список не містить дат, і SOME d, якщо дата d є найстарішою датою у списку.

2. Опис програмного коду

task.sml

```
(*1*)
fun is_older (date1: int* int* int, date2: int* int* int) =
    if (#1 date1) < (#1 date2) then
        true
    else if (#2 date1) < (#2 date2) andalso (#1 date1) = (#1 date2) then
        true
    else
        if (#3 date1) < (#3 date2) andalso (#1 date1) = (#1 date2) andalso (#2 date1) = (#2 date2) then
            true
        else
            false


    (*2*)
fun number_in_month (dates : (int*int*int) list, month : int) =
    if null dates
    then 0
    else
        if #2 (hd dates) = month then
            number_in_month(tl dates, month) + 1
        else
            number_in_month(tl dates, month);


    (*3*)
fun number_in_months (dates : (int*int*int) list, months : int list) =
    if null months then
        0
    else
        number_in_months(dates, tl months) + number_in_month(dates, hd months);


    (*4*)
fun dates_in_month (dates : (int*int*int) list, month : int) =
    if null dates then
        []
    else
        if #2 (hd dates) = month then
          (hd dates) :: dates_in_month(tl dates, month)
        else
           dates_in_month(tl dates, month);
```

```
42.
43.     (*5*)
44. fun dates_in_months (dates : (int*int*int) list, months : int list) =
45.     if null months then
46.         []
47.     else
48.        dates_in_month(dates, hd months) @ dates_in_months(dates, tl months);
49.

50.     (*6*)
51. fun get_nth(strings : string list, n : int) =
52.     if (null strings) then
53.         ""
54.     else
55.         if (n = 1) then
56.             hd strings
57.         else
58.             get_nth(tl strings, n-1);
59.

60.     (*7*)
61. fun date_to_string(date : int*int*int) =
62.     get_nth([
63.         ("January"), ("February"), ("March"), ("April"),
64.         ("May"), ("June"), ("July"), ("August"),
65.          ("September"), ("October"), ("November"), ("December")
66.     ],(#2 date)) ^ " " ^ Int.toString(#3 date) ^ ", " ^ Int.toString(#1
    date);
67.

68.     (*8*)
69. fun number_before_reaching_sum (sum : int, lis : int list) =
70.     if null lis then
71.         0
72.     else
73.         if sum - (hd lis) > 0 then
74.             number_before_reaching_sum(sum - hd lis, tl lis) + 1
75.         else
76.           0;
77.

78.     (*9*)
79. fun what_month (day : int) =
80.     if (day > 0) andalso (day < 366) then
81.         number_before_reaching_sum(day,[31, 28, 31, 30, 31, 30, 31, 31, 30,
    31, 30, 31]) + 1
82.     else
83.         0;
84.

85.
```

```
    (*10*)
fun month_range(day1 : int, day2 : int) =
    if (day1 > day2) then
        []
    else
        if (day1 = day2) then
          what_month(day1) :: []
        else
          what_month(day1) :: month_range(day1 + 1, day2);


    (*11*)
fun oldest_date (dates : (int*int*int) list) =
    if null dates then
        NONE
          else
              let fun oldest_nonempty (dates : (int*int*int) list) =
                  if null (tl dates) then
                      hd dates
                  else let val tl_ans = oldest_nonempty(tl dates)
                      in
                          if is_older(hd dates, tl_ans) then
                              hd dates
                          else
                              tl_ans
                      end
          in
              SOME (oldest_nonempty(dates))
          end

```

3. Тести програми

test.sml

```
use "task.sml";

    (*test 1 task*)
fun provided1_test1 () =
    let val date1 = (2002,10,15)
        val date2 = (2002,10,15)
    in
        is_older(date1,date2)
    end;

fun provided1_test2 () =
    let val date1 = (2000,9,15)
        val date2 = (2000,10,15)
    in
        is_older(date1,date2)
    end;

val res1_1 = provided1_test1();
val res1_2 = provided1_test2();
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
-
val res1_1 = false : bool

val res1_2 = true : bool
-
```

```
    (*test 2 task*)
fun provided2_test1 () =
    let val dates = [(2020,12,6), (2021, 5,10), (2003,7,18)]
        val month = 7
    in
        number_in_month(dates, month)
    end;

fun provided2_test2 () =
    let val dates = [(2002,12,1),(2002,11,10),(2002,12,10)]
        val month = 12
    in
        number_in_month(dates, month)
    end;

val res2_1 = provided2_test1();
val res2_2 = provided2_test2();
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
-
val res2_1 = 1 : int

val res2_2 = 2 : int
-
```

```
    (*test 3 task*)
fun provided3_test1 () =
    let val dates = [(2003, 1, 30), (2003, 1, 30), (2003, 2, 30), (2003, 3, 30), (2003, 2, 30)]
        val months = [1, 2]
    in
        number_in_months(dates, months)
    end;

fun provided3_test2 () =
    let val dates = [(2002,12,1),(2002,11,10),(2002,12,10)]
        val months = [3, 10]
    in
        number_in_months(dates, months)
    end;

val res3_1 = provided3_test1();
val res3_2 = provided3_test2();
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
val res3_1 = 4 : int

val res3_2 = 0 : int
```

```
    (*test 4 task*)
fun provided4_test1 () =
    let val dates = [(2003, 1, 30), (2003, 2, 28), (2003, 2, 20), (2003, 1, 25), (2003, 1, 11)]
        val month = 1
    in
        dates_in_month(dates, month)
    end;

fun provided4_test2 () =
    let val dates = [(1990,3,15), (1997,12,7), (1558,2,11)]
        val month = 3
    in
        dates_in_month(dates, month)
    end;

val res4_1 = provided4_test1();
val res4_2 = provided4_test2();
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
-
val res4_1 = [(2003,1,30),(2003,1,25),(2003,1,11)] : (int * int * int) list

val res4_2 = [(1990,3,15)] : (int * int * int) list
-
```

```
    (*test 5 task*)
fun provided5_test1 () =
    let val dates = [(2003, 1, 3), (2001, 3, 1), (1507, 7, 1), (2000, 1, 14), (1976, 3, 15)]
        val months = [1,3]
    in
        dates_in_months(dates, months)
    end;

fun provided5_test2 () =
    let val dates = [(2003, 1, 30), (2003, 2, 28), (2003, 3, 20), (2003, 4, 25), (2003, 5, 11)]
        val months = [1, 10, 3, 5]
    in
        dates_in_months(dates, months)
    end;

val res5_1 = provided5_test1();
val res5_1 = provided5_test2();
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
val res5_1 = [(2003,1,3),(2000,1,14),(2001,3,1),(1976,3,15)] :
  (int * int * int) list

val res5_1 = [(2003,1,30),(2003,3,20),(2003,5,11)] : (int * int * int) list
```

```
    (*test 6 task*)
fun provided6_test1 () =
    let val strings = ["abc1","def2","3gjk","4klm","5nprs"]
        val numb = 3
    in
        get_nth(strings, numb)
    end;

fun provided6_test2 () =
    let val strings = ["jock", "fire", "water"]
        val numb = 2
    in
        get_nth(strings, numb)
    end;

val res6_1 = provided6_test1();
val res6_2 = provided6_test2();
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
val res6_1 = "3gjk" : string

val res6_2 = "fire" : string
```

```
    (*test 7 task*)
fun provided7_test1 () =
    let val date = (2002, 1, 19)
    in
        date_to_string(date)
    end;

fun provided7_test2 () =
    let val date = (2000, 10, 9)
    in
        date_to_string(date)
    end;

val res7_1 = provided7_test1();
val res7_2 = provided7_test2();

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
val res7_1 = "January 19, 2002" : string

val res7_2 = "October 9, 2000" : string
-
```

```

    (*test 8 task*)
fun provided8_test1 () =
    let val sum = 100
        val list = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
    in
        number_before_reaching_sum(sum, list)
    end;

fun provided8_test2 () =
    let val sum = 31
        val list = [15, 15, 10, 10]
    in
        number_before_reaching_sum(sum, list)
    end;

val res8_1 = provided8_test1();
val res8_2 = provided8_test2();

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
-
val res8_1 = 3 : int

val res8_2 = 2 : int
-
```

```
160
161     (*test 9 task*)
162   fun provided9_test1 () =
163     let val day = 100
164     in
165         what_month(day)
166     end;
167
168   fun provided9_test2 () =
169     let val day = 365
170     in
171         what_month(day)
172     end;
173
174   val res9_1 = provided9_test1();
175   val res9_2 = provided9_test2();
176
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
val res9_1 = 4 : int

val res9_2 = 12 : int
```

```
179     (*test 10 task*)
180   fun provided10_test1 () =
181     let val day1 = 90
182         val day2 = 101
183     in
184         month_range(day1, day2)
185     end;
186
187   fun provided10_test2 () =
188     let val day1 = 58
189         val day2 = 73
190     in
191         month_range(day1, day2)
192     end;
193
194   val res10_1 = provided10_test1();
195   val res10_2 = provided10_test2();
196
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
val res10_1 = [3,4,4,4,4,4,4,4,4,4,4,4] : int list

val res10_2 = [2,2,3,3,3,3,3,3,3,3,3,3,3,3,3] : int list
```

```
199     (*test 11 task*)
200 fun provided11_test1 () =
201     let val dates = [(1800,10,20), (1799,10,10), (1801,10,10)]
202     in
203         valOf(oldest_date(dates))
204     end;
205
206 fun provided11_test2 () =
207     let val dates = [(2003, 1, 3), (2001, 3, 1), (1976, 7, 1), (2000, 1, 14), (1976, 3, 15)]
208     in
209         valOf(oldest_date(dates))
210     end;
211 val res11_1 =  provided11_test1();
212 val res11_2 =  provided11_test2();
213
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
-
val res11_1 = (1799,10,10) : int * int * int

val res11_2 = (1976,3,15) : int * int * int
-
```