НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ім. Ігоря Сікорського»
Кафедра інформатики та програмної інженерії

МЛЬТИПАРАДИГМЕННЕ ПРОГРАМУВАННЯ

Лабораторна робота №2

Виконав: Студентка гр. ІТ – 02
Скачкова Анастасія Дмитрівна

Перевірив: Очеретяний Олександр Костянтинович

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.)

Практична частина:

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.)

Лістинг програми:

```
fun is_older(tr:int*int*int, tr2:int*int*int) =
if (#1 tr) < (#1 tr2)
then true
else if ((#1 tr) = (#1 tr2)) andalso ((#2 tr) < (#2 tr2))
then true
else if ((#2 tr) = (#2 tr2)) andalso ((#3 tr) < (#3 tr2))
then true
else false;

val res = is_older((2002,3,33), (2002,3,33));
val res = is_older((1996,2,5), (1999,2,5));
```

Результат:

```
val is_older = fn : (int * int * int) * (int * int * int) -> bool

val res = false : bool

val res = true : bool
```

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

Лістинг програми:

```
fun number_in_month(xs:(int*int*int) list, month:int) =
```

```
if null xs
then 0
else if (month = (#2 (hd xs)))
then 1 + number_in_month(tl xs, month)
else number_in_month(tl xs, month);

val res_2 = number_in_month([(2002,3,33), (2002,3,33), (1996,2,5)], 3);
val res_2 = number_in_month([(1996,2,5), (1999,2,5), (1996,5,6), (1999,11,5),
(1996,10,5), (1999,5,5)], 2);
```

Результат:

```
val number_in_month = fn : (int * int * int) list * int -> int

val res_2 = 2 : int

val res_2 = 2 : int
```

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.

Лістинг програми:

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

val res_3 = number_in_months([],[]);
val res_3 = number_in_months([(2002,3,33), (2002,6,33), (1996,2,5)], [2, 3, 6]);
val res_3 = number_in_months([(1996,2,5), (1999,3,5), (1996,5,6), (1999,1,5),
(1996,10,5), (1999,6,5)], [1,2]);
```

Результат:

```
val number_in_months = fn : (int * int * int) list * int list -> int

val res_3 = 0 : int

val res_3 = 3 : int

val res_3 = 2 : int
```

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.

Лістинг програми:

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

val res_4 = dates_in_month([(2002,3,33), (2002,3,33), (1996,2,5)], 3);
val res_4 = dates_in_month([(1996,2,5), (1999,2,5), (1996,5,6), (1999,11,5),
(1996,10,5), (1999,2,5)], 2);
```

Результат:

```
val dates_in_month = fn :
  (int * int * int) list * int -> (int * int * int) list

val res_4 = [(2002,3,33),(2002,3,33)] : (int * int * int) list

val res_4 = [(1996,2,5),(1999,2,5),(1999,2,5)] : (int * int * int) list
```

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 (@).

Лістинг програми:

```
fun dates_in_months(xs: (int*int*int) list, months: int list) =
if null xs orelse null months
then []
else dates_in_month(xs, (hd months))@dates_in_months(xs, (tl months));

val res_5 = number_in_months([],[]);
val res_5 = number_in_months([(2002,3,33), (2002,6,33), (1996,2,5)], [2, 3, 6]);
val res_5 = number_in_months([(1996,2,5), (1999,3,5), (1996,5,6), (1999,1,5),
(1996,10,5), (1999,6,5)], [1,2]);
```

Результат:

```
val dates_in_months = fn :
  (int * int * int) list * int list -> (int * int * int) list

val res_5 = 0 : int

val res_5 = 3 : int

val res_5 = 2 : int
```

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.

Лістинг програми:

```
fun get_nth(text: string list, number: int) =
if null text
then "Element at this number doesn't exist"
else if number=1
then (hd text)
else get_nth((tl text), number-1);

val res_6 = get_nth(["str", "str2", "str3", "str4", "str5", "str6"], 4);
val res_6 = get_nth(["str", "str2", "str3", "str4", "str5", "str6"], 9);
val res_6 = get_nth([], 4);
```

Результат:

```
val get_nth = fn : string list * int -> string

val res_6 = "str4" : string

val res_6 = "Element at this number doesn't exist" : string

val res_6 = "Element at this number doesn't exist" : string
```

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.

Лістинг програми:

```
fun date_to_string(xs: (int*int*int)) =
let val months = ["January", "February", "March", "April", "May", "June", "July",
"August", "September", "October", "November", "December"]
```

```
in
get_nth(months,#2 xs)^" "^(Int.toString (#3 xs))^", "^(Int.toString (#1 xs))
end;

val res_7 = date_to_string((2022, 2, 28));
```

Результат:

```
val date_to_string = fn : int * int * int -> string

val res_7 = "February 28, 2022" : string
```

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.

Лістинг програми:

```
fun number_before_reaching_sum(sum: int, xs: int list) =
if null xs
then 0
else if sum - (hd xs) > 0
then 1 + number_before_reaching_sum(sum - (hd xs), tl xs)
else 0;

val res_8 = number_before_reaching_sum(10, [1, 2, 2, 4, 3]);
```

Результат:

```
val number_before_reaching_sum = fn : int * int list -> int

val res_8 = 4 : int
```

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.

Лістинг програми:

```
fun what_month(day: int) =
if day<1 orelse day>365
then 0
else let val days_of_months = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
in
number_before_reaching_sum(day, days_of_months)+1
end;

val res_9 = what_month(366);
```

```
val res_9 = what_month(31);
val res_9 = what_month(32);
```

Результат:

```
val what_month = fn : int -> int

val res_9 = 0 : int

val res_9 = 1 : int

val res_9 = 2 : int
```

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, ..., а mn – місяць day2. Зверніть увагу, що результат матиме довжину day2 - day1 + 1 або довжину 0, якщо day1>day2.

Лістинг програми:

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

val res_10 = month_range(22, 33);
val res_10 = month_range(364, 368);
```

Результат:

```
val month_range = fn : int * int -> int list

val res_10 = [1,1,1,1,1,1,1,1,1,1,2,2] : int list

val res_10 = [12,12,0,0,0] : int list
```

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.

Лістинг програми:

```
fun legacy(tr : (int*int*int) list) =
if null tr
then NONE
else let
fun max_nonempty (tr : (int*int*int) list) =
if null (tl tr)
then (hd tr)
else let val tl_ans = max_nonempty(tl tr)
```

```
in
if is_older(hd tr, tl_ans)
then tl_ans
else hd tr
end
in
SOME (max_nonempty tr)
end;

val res_11 = legacy([(1996,2,5), (1999,2,5), (1996,5,6), (1999,11,5), (1996,10,5),
(1999,5,5)]);
val res_11 = legacy([(1996,2,5), (2002,3,33), (2003,3,33)]);
val res_11 = legacy([]);
```

Результат:

```
val legacy = fn : (int * int * int) list -> (int * int * int) option

val res_11 = SOME (1999,11,5) : (int * int * int) option

val res_11 = SOME (2003,3,33) : (int * int * int) option

val res_11 = NONE : (int * int * int) option
```