

**CS29206 Systems Programming Laboratory**  
**Spring 2024**

**Lab Assignment: 1**

**Date: 18-Jan-2024**

---

**Multi-file Applications, Libraries, and Application Programs**

**The set ADT**

Let  $N$  be a fixed positive integer. Consider the universal set  $U_N = \{1, 2, \dots, N\}$ . Our set ADT deals with subsets of  $U_N$ . Let us call such a subset a number set (*numset*). The ADT should support the following functions.

**Initialize and destroy numsets**

$S = \text{setinit}(N)$	Initialize a numset $S$ to the empty subset of $U_N$
$S = \text{setrand}(N)$	Initialize a numset $S$ to a random subset of $U_N$
$S = \text{setrandsize}(N, t)$	Initialize a numset $S$ to a random subset of $U_N$ of size $t$
$S = \text{setdestroy}(S)$	Make $S$ an invalid set

**Basic set operations** (Here,  $S, T, A, B$  are of type *numset*, and  $t$  and  $i$  are integers)

$\text{setprint}(S)$	Print the elements of a numset $S$ (nothing to return)
$t = \text{setsize}(S)$	Return the size (number of elements) in the numset $S$ , that is, $t :=  S $
$T = \text{setaddelt}(S, i)$	Add the element $i$ to the numset $S$ , that is, $T := S \cup \{i\}$
$T = \text{setdelelt}(S, i)$	Remove the element $i$ from the numset $S$ , that is, $T := S - \{i\}$
$T = \text{setunion}(A, B)$	Set $T := A \cup B$
$T = \text{setintersection}(A, B)$	Set $T := A \cap B$

**Chocolates and coupons**

The FooBar Choco Company (FCC) sells 100gm chocolate packets, each containing one of  $N$  coupons numbered  $1, 2, \dots, N$ . If you can collect all of these  $N$  coupons and send those to the FCC Headquarters, you are guaranteed to receive a fabulous gift from FCC. In the manufacturing plants of FCC, one of these  $N$  coupons is uniformly randomly chosen for inserting in each 100gm FCC chocolate packet.

You need to figure out how many chocolate packets you need to purchase for collecting all of these  $N$  coupons so that you can claim the gift from FCC. Statistical analysis establishes that the expected number of purchases would be  $NH_N$  (where  $H_N$  is the  $N$ -th harmonic number).

**What you have to do**

● **Prepare a *numset* library**

Represent a *numset*  $S$  by a dynamically allocated *int* array of size  $N + 1$ .  $S[0]$  will store the size  $N$  of the universe. For  $i = 1, 2, \dots, N$ , the array entry  $S[i]$  will store 1 if  $i$  is a member of  $S$  or 0 otherwise.

Write a header file *settype.h* to define the data type *numset* and to declare the prototypes of the functions to initialize and destroy a *numset*. Write a C source file *settype.c* to implement these functions.

Write a header file *setops.h* and a C source file *setops.c* for the basic operations on *numsets* as given in the ADT specification above.

You must strictly stick to the input/output behavior of the ADT functions as mentioned above. These files should be compilable to a static library *libsetutils.a* or a shared (that is, dynamic) library *libsetutils.so*. There should not be any *main()* function in any of these two source files.

## ● Write an application program *chocolate.c*

The basic task of this program is to find how many chocolate packets need to be purchased until the FCC gift can be claimed. You must make use of the *setutils* library calls to achieve this. (No credits for doing it in any other manner.) Your C program *chocolate.c* starts by reading  $N$  from the user. This value of  $N$  will be used throughout the rest of the program.

The following pseudocode generates a set  $S$  of all types of coupons. Write a function *howmanypackets*( $N$ ) that implements this pseudocode using a *numset*  $S$ , and returns the number of iterations of the while loop.

```
Initialize  $S := \emptyset$ .
While ( $|S| < N$ ), repeat:
    Generate a random coupon  $c$  from  $\{1, 2, 3, \dots, N\}$ .
    Update  $S := S \cup \{c\}$ .
```

During the compilation of *chocolate.c*, you can define a flag `STAT_MODE` (**statistics mode**). If this flag is not defined (this mode is called the **interactive mode**), you make a verbose reporting of the working of a single invocation of *howmanypackets*() (see the Sample Output below to know what to print in this mode).

A single invocation of *howmanypackets*() would not give you a clear statistical idea. If you invoke the function sufficiently many times, then you expect the sample average to be close to the actual average  $NH_N$  (you do not need to compute this quantity). If the flag `STAT_MODE` is set (that is, in the statistics mode), you make 1000 calls of *howmanypackets*(), and report the average of the return values from these calls. Note that you start each purchase sequence with an empty collection of coupons, that is, you do not use the leftover coupons from one purchase sequence in later purchase sequences. Do not make a verbose output in the statistics mode (see the Sample Output below).

A random coupon  $c$  in the range  $1, 2, \dots, N$  can be obtained by the call  $1 + \text{rand}() \% N$ . Add the line `srand((unsigned int)time(NULL));` once at the beginning of *chocolate.c* to get different outputs in different runs. The functions *rand*() and *srand*() are declared in `<stdlib.h>`. The *time*() function needs `<time.h>`.

## ● Scripts

Write a shell script *runstatic* that contains the following commands.

1. Remove all the binary files (if any) from the previous run.
2. Compile individual source files for the library.
3. Build the static library from the object files.
4. Compile the application program *chocolate.c* in the interactive mode.
5. Run the compiled program once.
6. Re-compile the application program *chocolate.c* in the statistics mode.
7. Run the compiled program once.

Add execute permission (for all) to the script *runstatic*, and run it in your terminal as `./runstatic`.

Repeat Steps 1–7, but you now create a dynamic *setutils* library (Step 3). Name your script file in this case as *rundynamic*. Add execute permission to *rundynamic*, and run it in your terminal as `./rundynamic`.

## ● Combine all you have done

Remove all the binary files, and submit a zip file *LAI.zip* containing only the following files:

`chocolate.c`    `rundynamic*`    `runstatic*`    `setops.c`    `setops.h`    `settype.c`    `settype.h`

Here, \* is not part of a filename, it instead implies an executable file.

## ● Submit

Submit the single file *LAI.zip* to the CSE Moodle server. Please be careful about the submission deadline. The Moodle server time is displayed near the top left corner of the webpage.

## Sample Output

```
$. /runstatic

Number of coupons (N): 10
+++ Interactive mode
Packet 1 has coupon 8, Available coupons: {8}
Packet 2 has coupon 1, Available coupons: {1, 8}
Packet 3 has coupon 3, Available coupons: {1, 3, 8}
Packet 4 has coupon 8, Available coupons: {1, 3, 8}
Packet 5 has coupon 4, Available coupons: {1, 3, 4, 8}
Packet 6 has coupon 8, Available coupons: {1, 3, 4, 8}
Packet 7 has coupon 4, Available coupons: {1, 3, 4, 8}
Packet 8 has coupon 9, Available coupons: {1, 3, 4, 8, 9}
Packet 9 has coupon 1, Available coupons: {1, 3, 4, 8, 9}
Packet 10 has coupon 9, Available coupons: {1, 3, 4, 8, 9}
Packet 11 has coupon 3, Available coupons: {1, 3, 4, 8, 9}
Packet 12 has coupon 6, Available coupons: {1, 3, 4, 6, 8, 9}
Packet 13 has coupon 9, Available coupons: {1, 3, 4, 6, 8, 9}
Packet 14 has coupon 3, Available coupons: {1, 3, 4, 6, 8, 9}
Packet 15 has coupon 3, Available coupons: {1, 3, 4, 6, 8, 9}
Packet 16 has coupon 2, Available coupons: {1, 2, 3, 4, 6, 8, 9}
Packet 17 has coupon 5, Available coupons: {1, 2, 3, 4, 5, 6, 8, 9}
Packet 18 has coupon 3, Available coupons: {1, 2, 3, 4, 5, 6, 8, 9}
Packet 19 has coupon 10, Available coupons: {1, 2, 3, 4, 5, 6, 8, 9, 10}
Packet 20 has coupon 6, Available coupons: {1, 2, 3, 4, 5, 6, 8, 9, 10}
Packet 21 has coupon 3, Available coupons: {1, 2, 3, 4, 5, 6, 8, 9, 10}
Packet 22 has coupon 9, Available coupons: {1, 2, 3, 4, 5, 6, 8, 9, 10}
Packet 23 has coupon 5, Available coupons: {1, 2, 3, 4, 5, 6, 8, 9, 10}
Packet 24 has coupon 8, Available coupons: {1, 2, 3, 4, 5, 6, 8, 9, 10}
Packet 25 has coupon 7, Available coupons: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
--- 25 packets were bought

Number of coupons (N): 10
+++ Statistics mode
--- Average number of packets to buy = 29.131000
$. /rundynamic

Number of coupons (N): 12
+++ Interactive mode
Packet 1 has coupon 7, Available coupons: {7}
Packet 2 has coupon 3, Available coupons: {3, 7}
Packet 3 has coupon 2, Available coupons: {2, 3, 7}
Packet 4 has coupon 2, Available coupons: {2, 3, 7}
Packet 5 has coupon 12, Available coupons: {2, 3, 7, 12}
Packet 6 has coupon 7, Available coupons: {2, 3, 7, 12}
Packet 7 has coupon 10, Available coupons: {2, 3, 7, 10, 12}
Packet 8 has coupon 9, Available coupons: {2, 3, 7, 9, 10, 12}
Packet 9 has coupon 11, Available coupons: {2, 3, 7, 9, 10, 11, 12}
Packet 10 has coupon 8, Available coupons: {2, 3, 7, 8, 9, 10, 11, 12}
Packet 11 has coupon 8, Available coupons: {2, 3, 7, 8, 9, 10, 11, 12}
Packet 12 has coupon 2, Available coupons: {2, 3, 7, 8, 9, 10, 11, 12}
Packet 13 has coupon 4, Available coupons: {2, 3, 4, 7, 8, 9, 10, 11, 12}
Packet 14 has coupon 3, Available coupons: {2, 3, 4, 7, 8, 9, 10, 11, 12}
Packet 15 has coupon 2, Available coupons: {2, 3, 4, 7, 8, 9, 10, 11, 12}
Packet 16 has coupon 12, Available coupons: {2, 3, 4, 7, 8, 9, 10, 11, 12}
Packet 17 has coupon 8, Available coupons: {2, 3, 4, 7, 8, 9, 10, 11, 12}
Packet 18 has coupon 3, Available coupons: {2, 3, 4, 7, 8, 9, 10, 11, 12}
Packet 19 has coupon 5, Available coupons: {2, 3, 4, 5, 7, 8, 9, 10, 11, 12}
Packet 20 has coupon 2, Available coupons: {2, 3, 4, 5, 7, 8, 9, 10, 11, 12}
Packet 21 has coupon 3, Available coupons: {2, 3, 4, 5, 7, 8, 9, 10, 11, 12}
Packet 22 has coupon 11, Available coupons: {2, 3, 4, 5, 7, 8, 9, 10, 11, 12}
Packet 23 has coupon 11, Available coupons: {2, 3, 4, 5, 7, 8, 9, 10, 11, 12}
Packet 24 has coupon 7, Available coupons: {2, 3, 4, 5, 7, 8, 9, 10, 11, 12}
Packet 25 has coupon 10, Available coupons: {2, 3, 4, 5, 7, 8, 9, 10, 11, 12}
Packet 26 has coupon 5, Available coupons: {2, 3, 4, 5, 7, 8, 9, 10, 11, 12}
Packet 27 has coupon 9, Available coupons: {2, 3, 4, 5, 7, 8, 9, 10, 11, 12}
Packet 28 has coupon 3, Available coupons: {2, 3, 4, 5, 7, 8, 9, 10, 11, 12}
Packet 29 has coupon 4, Available coupons: {2, 3, 4, 5, 7, 8, 9, 10, 11, 12}
Packet 30 has coupon 7, Available coupons: {2, 3, 4, 5, 7, 8, 9, 10, 11, 12}
Packet 31 has coupon 11, Available coupons: {2, 3, 4, 5, 7, 8, 9, 10, 11, 12}
Packet 32 has coupon 3, Available coupons: {2, 3, 4, 5, 7, 8, 9, 10, 11, 12}
Packet 33 has coupon 9, Available coupons: {2, 3, 4, 5, 7, 8, 9, 10, 11, 12}
Packet 34 has coupon 4, Available coupons: {2, 3, 4, 5, 7, 8, 9, 10, 11, 12}
Packet 35 has coupon 8, Available coupons: {2, 3, 4, 5, 7, 8, 9, 10, 11, 12}
Packet 36 has coupon 12, Available coupons: {2, 3, 4, 5, 7, 8, 9, 10, 11, 12}
Packet 37 has coupon 11, Available coupons: {2, 3, 4, 5, 7, 8, 9, 10, 11, 12}
Packet 38 has coupon 5, Available coupons: {2, 3, 4, 5, 7, 8, 9, 10, 11, 12}
Packet 39 has coupon 1, Available coupons: {1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12}
Packet 40 has coupon 9, Available coupons: {1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12}
Packet 41 has coupon 4, Available coupons: {1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12}
Packet 42 has coupon 8, Available coupons: {1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12}
Packet 43 has coupon 3, Available coupons: {1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12}
Packet 44 has coupon 7, Available coupons: {1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12}
Packet 45 has coupon 10, Available coupons: {1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12}
Packet 46 has coupon 4, Available coupons: {1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12}
Packet 47 has coupon 11, Available coupons: {1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12}
Packet 48 has coupon 6, Available coupons: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}
--- 48 packets were bought

Number of coupons (N): 12
+++ Statistics mode
--- Average number of packets to buy = 37.190000
$
```