CSE 331 - Project 1

Ozan Şelte - 161044061

Usage Info:

The input must be non-negative integers with separated by spaces or newlines in a text file and **the file must be ended with a newline character.** The filename can be changed on the .data segment and the size of the file cannot be larger than 4096 bytes. This is the only limitation.

The file's first line is the set X in the assignment, every other line is considered as a new subset of the X. Every element of every subset must also be inside the X.

The program works dynamically. It gets the count of numbers, the count of subsets and the size of subsets. This count might be anything. **The only limitation is the input file's size.**

Examples:

X: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
S0: 1 3 5 7 8 9 10 11 12
S1: 2 4 6
S2: 1 4 8 3
S3: 10 11 12 13 14
S4: 12 13 14 15 16
S5: 14 15 16 17 18
S6: 19 20

```
--- --- --- --- ---
Beginning , Uncovered Set:  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 , Uncovered Count:  20
Selected Subset:  0 , Covered Count:  9 , Uncovered Set:  2 4 6 13 14 15 16 17 18 19 20 , Uncovered Count:  11
Selected Subset:  5 , Covered Count:  5 , Uncovered Set:  2 4 6 13 19 20 , Uncovered Count:  6
Selected Subset:  1 , Covered Count:  3 , Uncovered Set:  13 19 20 , Uncovered Count:  3
Selected Subset:  6 , Covered Count:  2 , Uncovered Set:  13 , Uncovered Count:  1
Selected Subset:  3 , Covered Count:  1 , Uncovered Set:  , Uncovered Count:  0

END
--- --- --- --- ---
```

If we remove 20 from S6, X cannot be fully covered with these subsets and the output will change.

```
--- --- --- --- ---
Beginning , Uncovered Set:  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 , Uncovered Count:  20
Selected Subset:  0 , Covered Count:  9 , Uncovered Set:  2 4 6 13 14 15 16 17 18 19 20 , Uncovered Count:  11
Selected Subset:  5 , Covered Count:  5 , Uncovered Set:  2 4 6 13 19 20 , Uncovered Count:  6
Selected Subset:  1 , Covered Count:  3 , Uncovered Set:  13 19 20 , Uncovered Count:  3
Selected Subset:  3 , Covered Count:  1 , Uncovered Set:  19 20 , Uncovered Count:  2
Selected Subset:  6 , Covered Count:  1 , Uncovered Set:  20 , Uncovered Count:  1

END
--- --- --- --- ---
```

Macros:

print_newline : Prints newline character to the console.
print_space : Prints space character to the console.
print_int : Prints integer with a newline character to the console.
print_intsp : Prints integer with a space character to the console.
print_str : Prints string with a newline character to the console.
print_strsp : Prints string with a space character to the console.
get_bytes : Gets a number of word spaces for a given number from the heap.
get_words : Gets a number of word spaces for a given number from the heap.
chr2dec : Calculates an integer with an integer and a character. If parameters are 1 and '8'; it multiplies 1 with 10, adds it the '8'(56) and subtract 48.
iter_idx : Iterates two registers. If the first register is counter and the other one is an address, it adds 1 to the first one and 4 to the second one.
store_sregs : Stores RA register and S registers to the stack with SP register.
load_sregs : Loads RA register and S registers from the stack with SP register.

Data:

ncnt : X's number count.
scnt : Set count.
nums : Beginning address of X's numbers at heap.
chck : Beginning address of check bytes. 0/covered, 1/uncovered numbers.
sets : Beginning address of sets' addresses. Stores an address for each set.
name : Name of the input file. It can be changed.
fbuf : Buffer which stores file's content. Size can be changed.

Functions:

begin : Main function. Organizes the arguments and the return values for other functions. Coordinates selecting a subset and clearing covered values with a loop.
openfile : Opens a file and returns the file descriptor.
closefile : Closes the given file.
exit : Terminates program.
readfile : Reads all contents of the file and stores at **fbuf**.
countncnt : Counts and returns the numbers count of X.
countscnt : Counts and returns the sets count.
initheaps : Initializes three different arrays from the heap and sets every element of **chck** to 1. 1) **nums**: ncnt * byte 2) **sets**: scnt * word 3) **chck**: ncnt * byte
getnums : Writes X's numbers to **nums**.
getsets : Initializes arrays for every set from the heap and writes numbers' inside them.
count_ones : Counts and returns uncovered elements count.
select_subset : Selects and returns the most covering subset's score and address addres. If a number's index is 1 in subset and 1 in **chck**, it takes a score.
clear_checks : Clears covered number's indexes from **chck**. Takes selected subset's beginning address as an argument.
get_index : Return a number's index in **nums**. Takes a number as an argument.