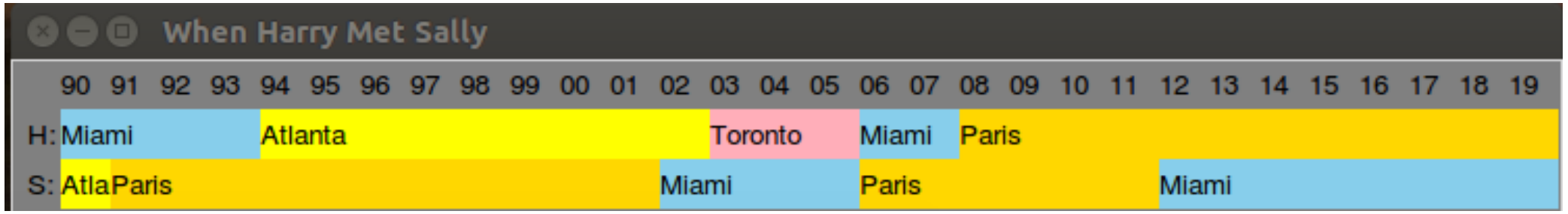
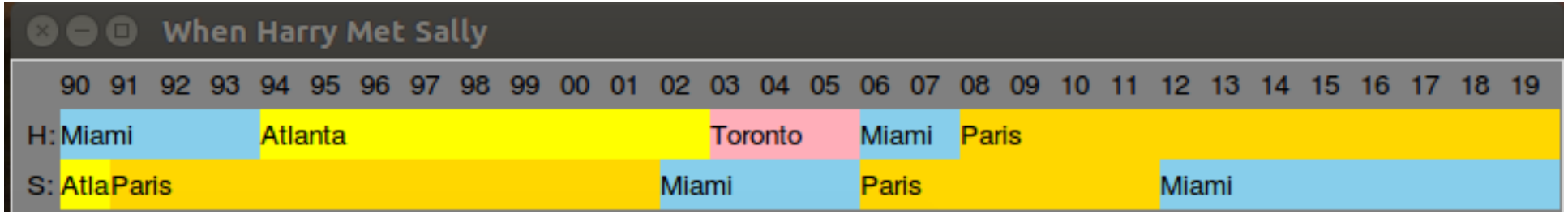
HW2: When Harry Met Sally



Topics:

- The puzzle your program must solve
- Command line arguments
- File I/O
- #define and enum
- swi interface for this assignment
- Oracle
- Creating test cases

The Puzzle: Find Earliest Time H&S are in Same City and Report the Year



city code for Miami

HarryTimeline[] = {4, 4, 9, 3, 3, 8, 2, 4, 12, 2};

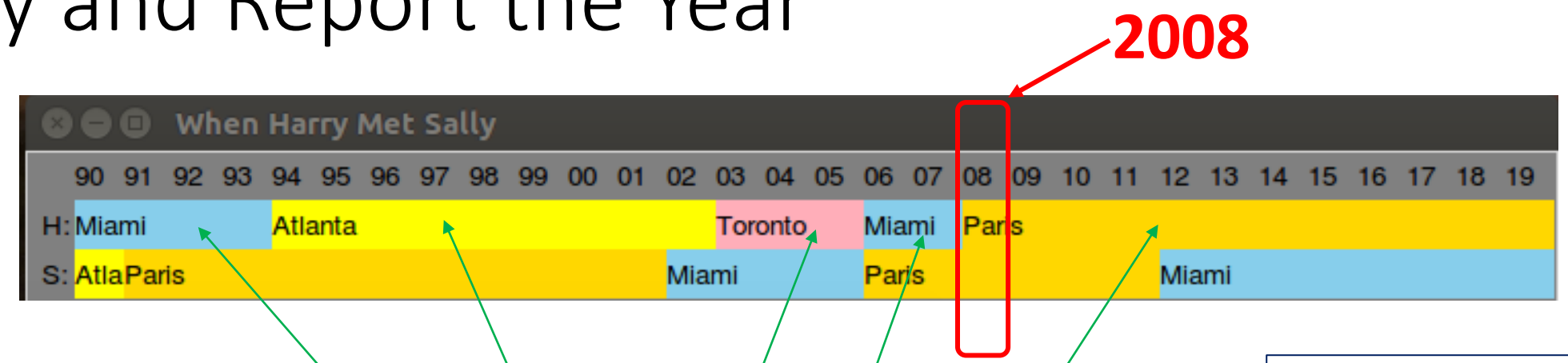
city code for Atlanta

SallyTimeline[] = {1, 3, 11, 2, 4, 4, 6, 2, 8, 4};

Sally lived in Paris (city code 2) for 11 years

First year is always 1990
City codes are in shell code

The Puzzle: Find Earliest Time H&S are in Same City and Report the Year



0 <= city code <= 9

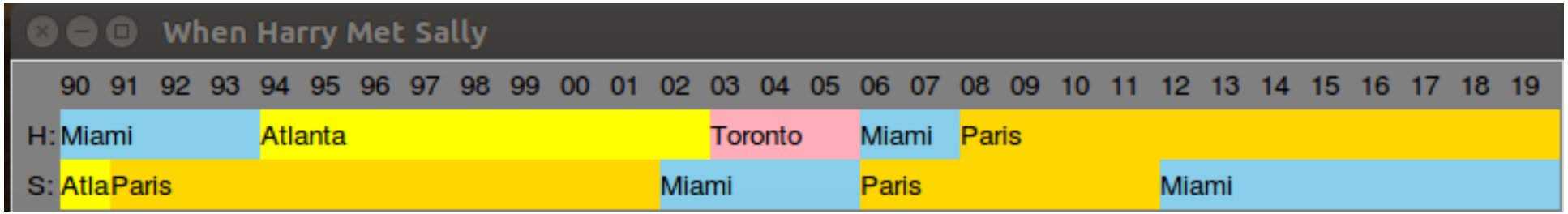
HarryTimeline[] = {4, 4, 9, 3, 3, 8, 2, 4, 12, 2};

SallyTimeline[] = {1, 3, 11, 2, 4, 4, 6, 2, 8, 4};

always 10 values (exactly 5 moves), in chronological order

Either consider the moves to be on Jan. 1 (or think of this as being where they lived most of the year)

HW2: Where Harry Met Sally



Topics:

- ✓ The puzzle your program must solve
- → Command line arguments
- File I/O
- #define and enum

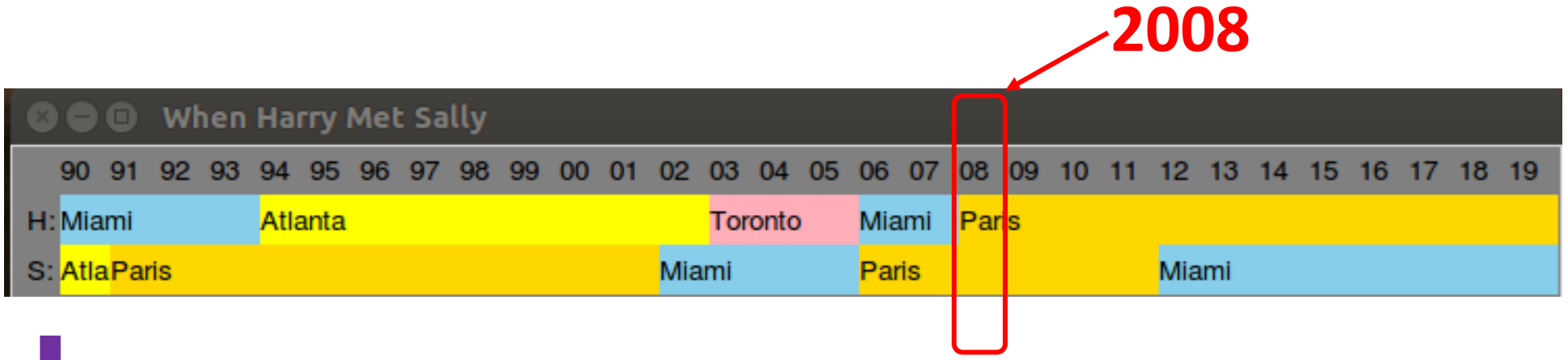
HW2-1.c

- swi interface for this assignment
- Oracle

HW2-2.asm

- Creating test cases

HW2-1: C solution takes a testcase file as input.



test2008.txt → HW2-1.c → Earliest year in common city: 2008

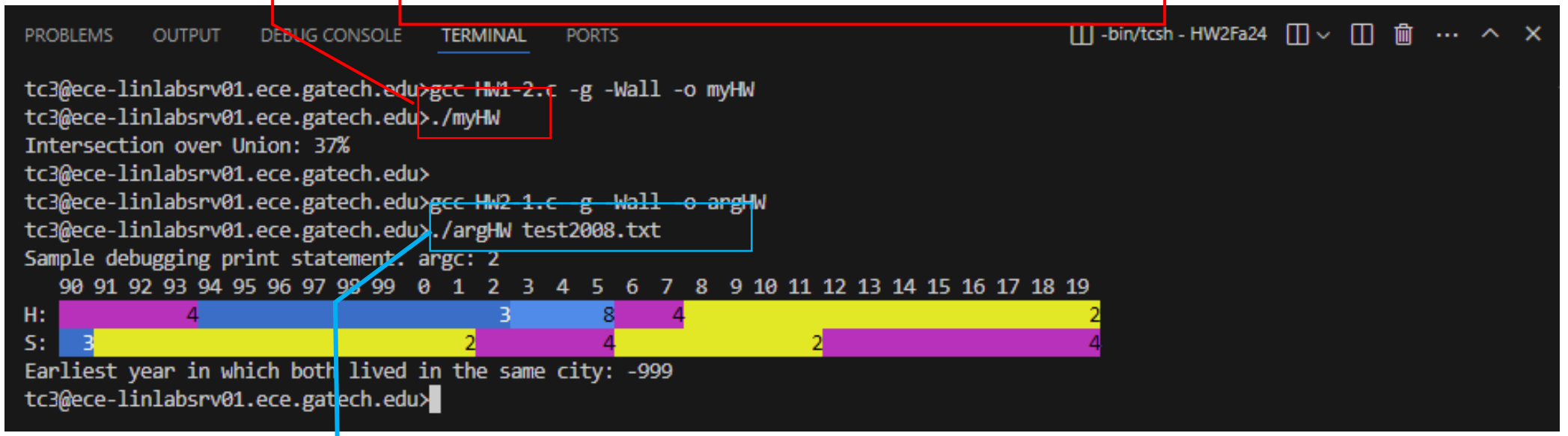
The answer to each test is in the filename!

Running C program: command-line arguments

Earlier assignments: Compile C file,

Run executable: calls main function

```
main()   // no arguments
```



NEW! Calls main function

with argument (test2008.txt)

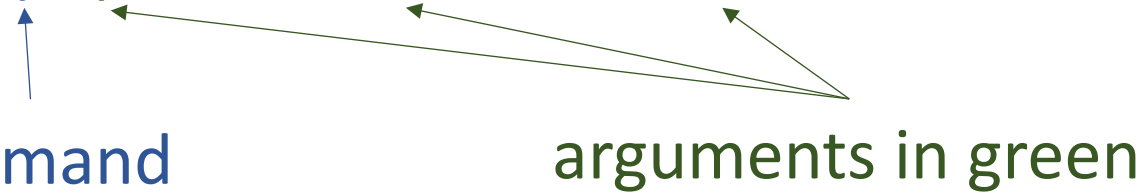
```
main(??)   // how?
```

Main: able to take variable number of inputs

linuxprompt$ ./argHW test2008.txt

linuxprompt$ ./ImageProgram 640 480

linuxprompt$ cp -p HW2-1-shell.c HW2-1.c



HOW?

```
33  */
34
35  #include <stdio.h>
36  #include <stdlib.h>
37
38  #define DEBUG 1 // RESET THIS TO 0 BEFORE SUBMITTING YOUR CODE
39
40  /* City IDs used in timelines. */
41  enum Cities{ London, Boston, Paris, Atlanta, Miami,
42               Tokyo, Metz, Seoul, Toronto, Austin };
43
44  int main(int argc, char *argv[]) {
45      int  HarryTimeline[10];
46      int  SallyTimeline[10];
47      int  NumNums, Year;
48      int  Load_Mem(char *, int *, int *);
49      void Print_Timelines(int *, int *);
50
51      if (argc != 2) {
52          printf("usage: ./HW2-1 valuefile\n");
53          exit(1);
54      }
55      NumNums = Load_Mem(argv[1], HarryTimeline, SallyTimeline);
56      if (NumNums != 20) {
57          printf("valuefiles must contain 20 entries\n");
```

argc: number of arguments

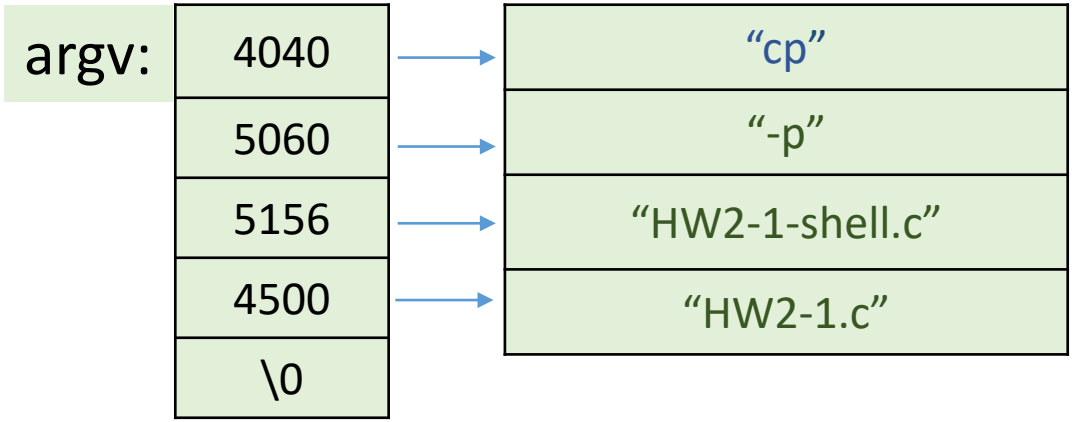
argv: array of base addresses of char arrays (strings)

Different example – Linux cp command

linuxprompt$ cp -p HW2-1-shell.c HW2-1.c

argc: number of arguments: 4 (includes cp)

argv: array of base addresses of char arrays (array of pointers to strings)

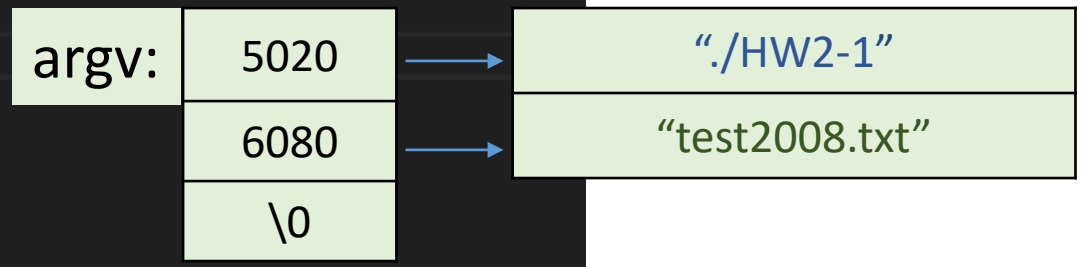


```
33  */
34
35  #include <stdio.h>
36  #include <stdlib.h>
37
38  #define DEBUG 1 // RESET THIS TO 0 BEFORE SUBMITT
39
40  /* City IDs used in timelines. */
41  enum Cities{ London, Boston, Paris, Atlanta, Miami,
42               Tokyo, Metz, Seoul, Toronto, Austin };
43
44  int main(int argc, char *argv[]) {
45    int  HarryTimeline[10];
46    int  SallyTimeline[10];
47    int  NumNums, Year;
48    int  Load_Mem(char *, int *, int *);
49    void Print_Timelines(int *, int *);
50
51    if (argc != 2) {
52      printf("usage: ./HW2-1 valuefile\n");
53      exit(1);
54    }
55    NumNums = Load_Mem(argv[1], HarryTimeline, SallyTimeline);
56    if (NumNums != 20) {
57      printf("valuefiles must contain 20 entries\n");
```

For HW2-1:
linuxprompt$ ./HW2-1 test2008.txt

argc: number of arguments: 2

argv: array of base addresses of char arrays (array of pointers to strings)

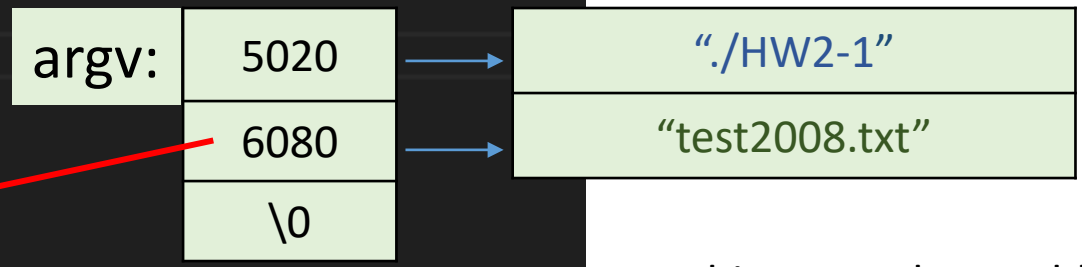


```
33  */
34
35  #include <stdio.h>
36  #include <stdlib.h>
37
38  #define DEBUG 1 // RESET THIS TO 0 BEFORE SUBMITT
39
40  /* City IDs used in timelines. */
41  enum Cities{ London, Boston, Paris, Atlanta, Miami,
42               Tokyo, Metz, Seoul, Toronto, Austin };
43
44  int main(int argc, char *argv[]) {
45    int  HarryTimeline[10];
46    int  SallyTimeline[10];
47    int  NumNums, Year;
48    int  Load_Mem(char *, int *, int *);
49    void Print_Timelines(int *, int *);
50
51    if (argc != 2) {
52      printf("usage: ./HW2-1 valuefile\n");
53      exit(1);
54    }
55    NumNums = Load_Mem(argv[1], HarryTimeline, SallyTimeline);
56    if (NumNums != 20) {
57      printf("valuefiles must contain 20 entries\n");
```

For HW2-1:
linuxprompt$ ./HW2-1 test2008.txt

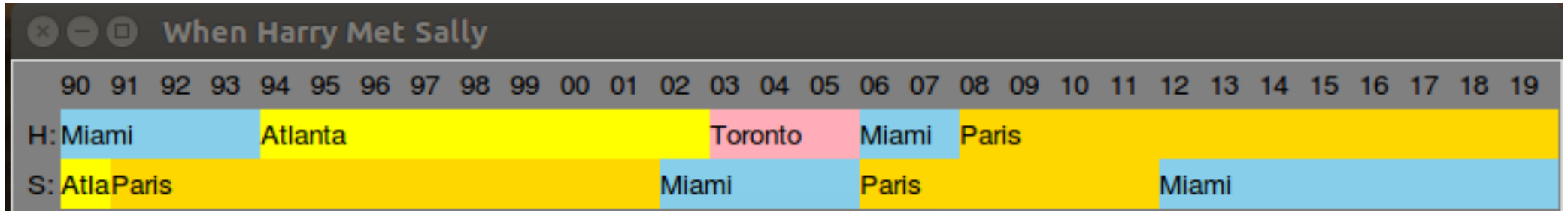
argc: number of arguments: 2

argv: array of base addresses of char arrays (array of pointers to strings)



This passes base address of testcase filename to Load_Mem.

Load_Mem: helper function in HW2-1-shell code



test2008.txt

```
5524:        4
5528:        4
5532:        9
5536:        3
5540:        3
5544:        8
...
```

Number in this column is meaningless here

HW2-1.c

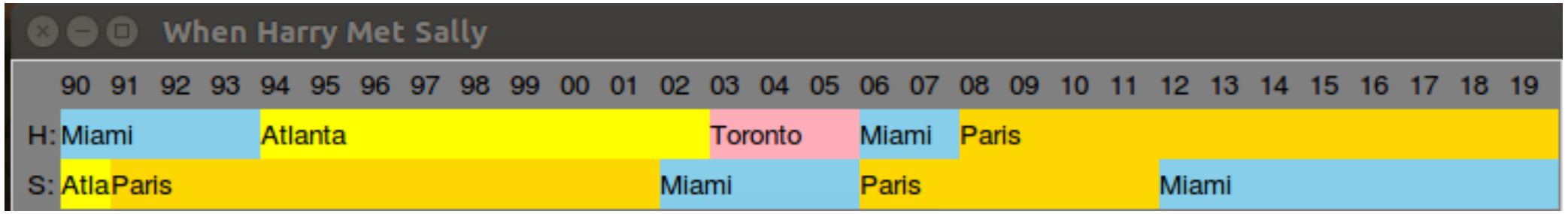
`Load_Mem`

HarryTimeline[] = {4, 4, 9, 3, 3, 8, 2, 4, 12, 2};
SallyTimeline[] = {1, 3, 11, 2, 4, 4, 6, 2, 8, 4};

`your code…`

Earliest common year: 2008

HW2: Where Harry Met Sally



Topics:

- ✓ The puzzle your program must solve
- ✓ Command line arguments
- → File I/O (Load_Mem example)
- #define and enum

HW2-1.c

- swi interface for this assignment
- Oracle

HW2-2.asm

- Creating test cases

File I/O in Load_Mem Function

```
81  /* This routine loads in up to 20 newline delimited integers from
82  a named file in the local directory. The values are placed in the
83  passed integer array. The number of input integers is returned. */
84
85  int Load_Mem(char *InputFileName, int IntArray1[], int IntArray2[]) {
86      int  N, Addr, Value, NumVals;
87      FILE *FP;
88
89      FP = fopen(InputFileName, "r");
90      if (FP == NULL) {
91        printf("%s could not be opened; check the filename\n", InputFileName);
92        return 0;
93      } else {
94        for (N=0; N < 20; N++) {
95          NumVals = fscanf(FP, "%d: %d", &Addr, &Value);
96          if (NumVals == 2)
97      if (N < 10)
98        IntArray1[N] = Value;
99      else
100       IntArray2[N-10] = Value;
          else
        break;
        }
        fclose(FP);
        return N;
    }
```

Mode (“r”/ “w” / “a”: read/write/append)

Read in pair of numbers in string of form “n: m” and store them in `Addr` and `Value`.

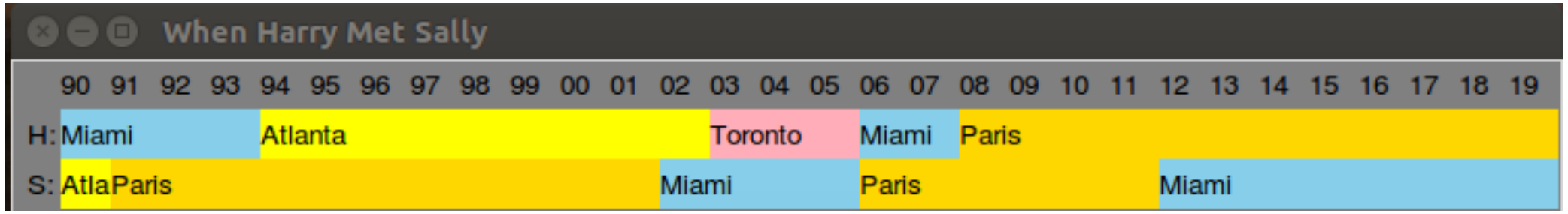
- returns number of successfully read values
- may return EOF (= -1)

`break` stops loop if current line read from file does not have exactly 2 ints

When file no longer needed, it should be closed.

stdio.h: defines C run-time library functions (fopen, fscanf, fclose)

HW2: Where Harry Met Sally



Topics:

- ✓ The puzzle your program must solve
- ✓ Command line arguments
- ✓ File I/O (Load_Mem example)
- → #define and enum

HW2-1.c

- swi interface for this assignment
- Oracle

HW2-2.asm

- Creating test cases

#define directive defines macro: replace “DEBUG” with 0 everywhere.

```c
#include <stdio.h>
#include <stdlib.h>

#define DEBUG 0 // RESET THIS TO 0 BEFORE SUBMITTING YOUR CODE

/* City IDs used in timelines. */
enum Cities{ London, Boston, Paris, Atlanta, Miami,
             Tokyo, Metz, Seoul, Toronto, Austin };

int main(int argc, char *argv[]) {
   int  HarryTimeline[10];
   int  SallyTimeline[10];
   ...
   if (DEBUG){
     printf("Sample debugging print statement. argc: %d \n", argc);
     printf("London: %d\n", London);
     printf("Boston: %d\n", Boston);
     for(i = 1; i<10; i=i+2){
       if (HarryTimeline[i] == Atlanta)
         printf("Harry visited Atlanta!\n");
     }
   }
```

0

Not executed – easy way to turn on/off debugging print statements.

Instead of laboriously commenting them out, simply set DEBUG to 0.
For running verbose, set DEBUG to 1. ***You must reset DEBUG to 0 before submitting your code!***

enum assigns names to integers

```c
#include <stdio.h>
#include <stdlib.h>

#define DEBUG 1 // RESET THIS TO 0 BEFORE SUBMITTING YOUR CODE

/* City IDs used in timelines. */
enum Cities{ London, Boston, Paris, Atlanta, Miami,
             Tokyo, Metz, Seoul, Toronto, Austin };

int main(int argc, char *argv[]) {
   int  HarryTimeline[10];
   int  SallyTimeline[10];
   ...
   if (DEBUG){
     printf("Sample debugging print statement. argc: %d \n", argc);
     printf("London: %d\n", London);
     printf("Boston: %d\n", Boston);
     for(i = 1; i<10; i=i+2){
       if (HarryTimeline[i] == Atlanta)
         printf("Harry visited Atlanta!\n");
     }
   }
```

0 1 2 3 4 ...

1

0

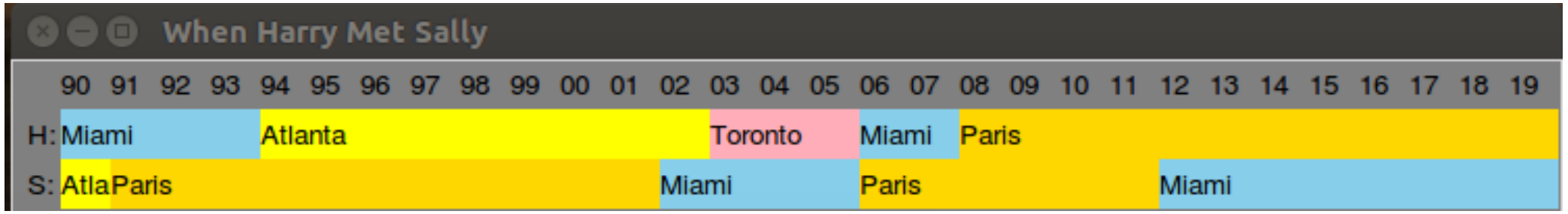
1

3

```
linda@Sassafras:/mnt/c/Users/Linda Wills/Documents/classes$ ./HW2-1 test2.txt
Sample debugging print statement. argc: 2
London: 0
Boston: 1
Harry visited Atlanta!

linda@Sassafras:/mnt/c/Users/Linda Wills/Documents/classes$
```

HW2: Where Harry Met Sally



Topics:

- ✓ The puzzle your program must solve
- ✓ Command line arguments
- ✓ File I/O (Load_Mem example)
- ✓ #define and enum

HW2-1.c

- → swi interface for this assignment
- Oracle

HW2-2.asm

- Creating test cases

Software Interrupt (swi) Instructions for HW2

HW2-2-shell.asm

```
.data
Harry:   .alloc  10. # happens to be at 5156 here
Sally:   .alloc  10

.text
WhenMet:addi   $1, $0, Harry      # set memory base
        swi      597              # create timelines
         # your code goes here.

         addi     $2, $0, 0       # guess 0
        swi      587              # give answer
        jr       $31              # return to caller
```

swi 597

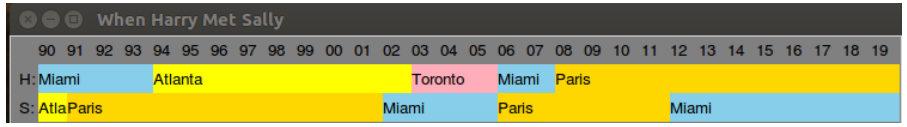
swi 587

Request timeline puzzle

- input: base address of space you allocated for the timelines in register $1
- output: Misasim will store timelines in memory starting at that base address

Memory Pane

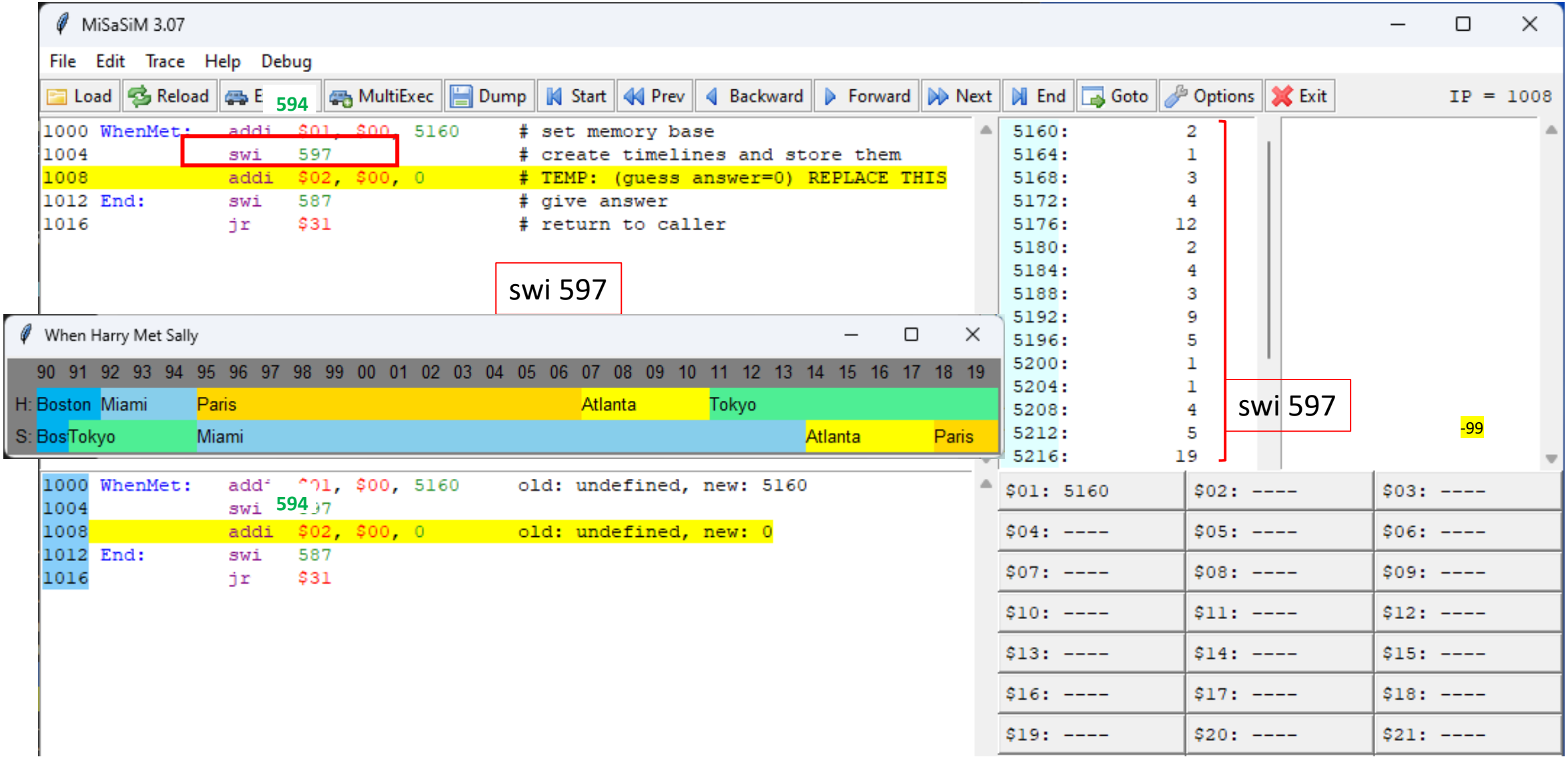
```
5160:        4
5164:        4
5168:        9
5172:        3
5176:        3
...
```

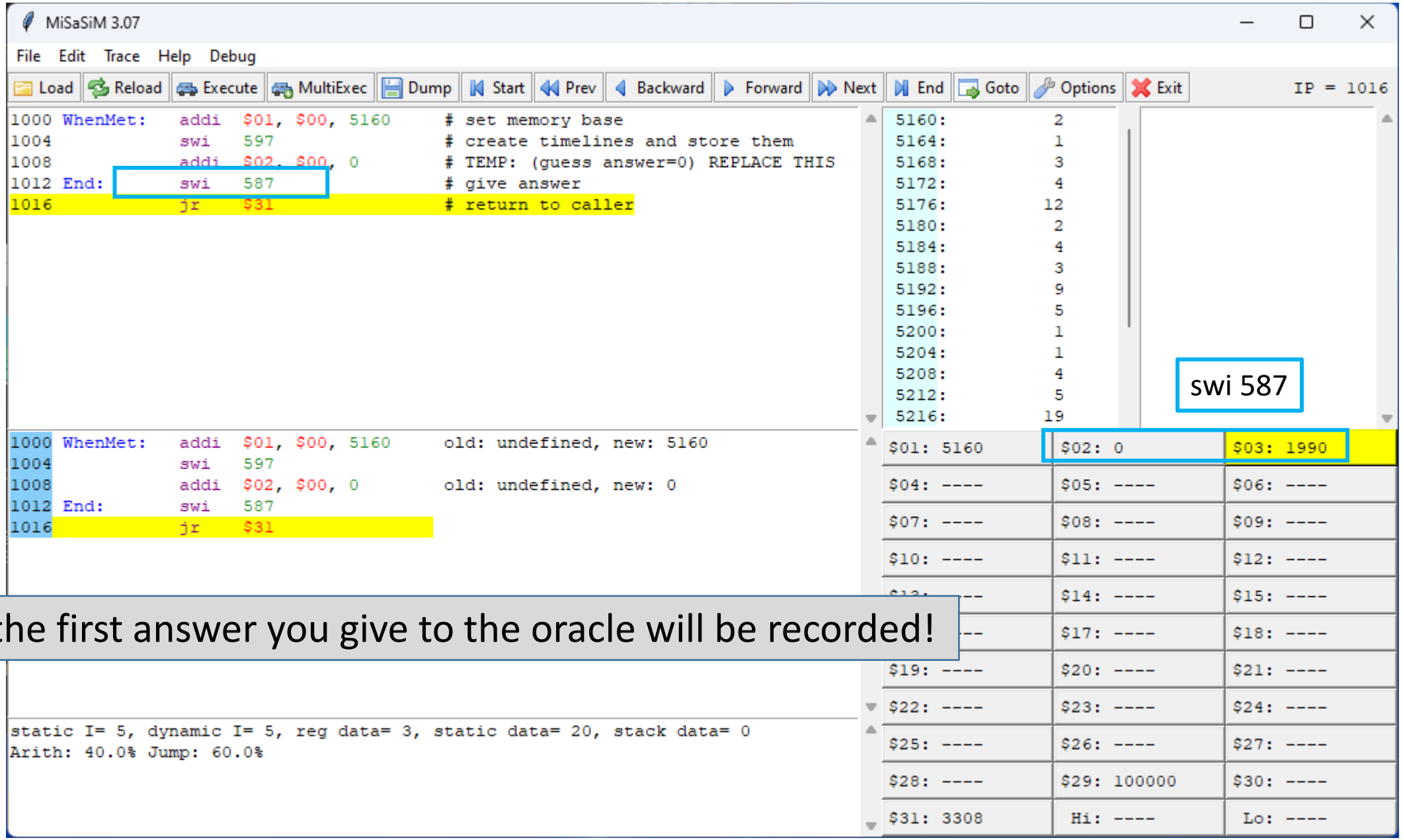


Report the answer your program computed

- input: your answer in register $2
- output: Oracle's answer in register $3

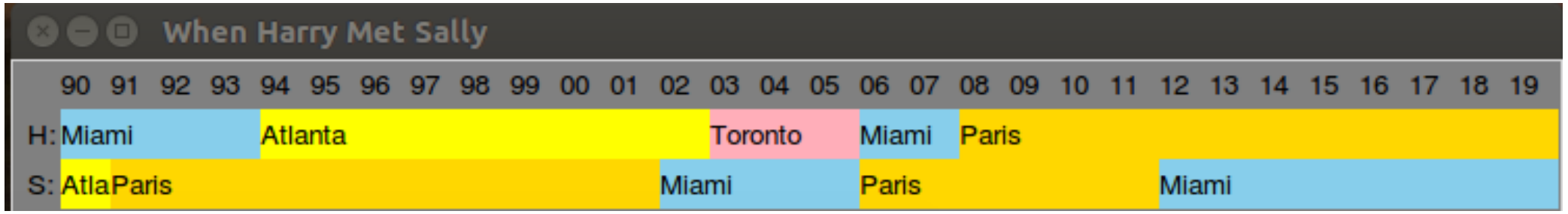
Now, this column shows the actual location used by Misasim (starting with 5160, the number passed in to swi 597, because Harry=5160 on this run.


MiSaSiM 3.07
File Edit Trace Help Debug
Load Reload 594 MultiExec Dump Start Prev Backward Forward Next End Goto Options Exit
IP = 1008
1000 WhenMet: addi $01, $00, 5160 # set memory base
1004 swi 597 # create timelines and store them
1008 addi $02, $00, 0 # TEMP: (guess answer=0) REPLACE THIS
1012 End: swi 587 # give answer
1016 jr $31 # return to caller
swi 597
When Harry Met Sally
90 91 92 93 94 95 96 97 98 99 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19
H: Boston Miami Paris Atlanta Tokyo
S: Bos Tokyo Miami Atlanta Paris
5160: 2
5164: 1
5168: 3
5172: 4
5176: 12
5180: 2
5184: 4
5188: 3
5192: 9
5196: 5
5200: 1
5204: 1
5208: 4
5212: 5
5216: 19
swi 597
-99
1000 WhenMet: addi $01, $00, 5160 old: undefined, new: 5160
1004 swi 597
1008 addi $02, $00, 0 old: undefined, new: 0
1012 End: swi 587
1016 jr $31
$01: 5160
$02: ----
$03: ----
$04: ----
$05: ----
$06: ----
$07: ----
$08: ----
$09: ----
$10: ----
$11: ----
$12: ----
$13: ----
$14: ----
$15: ----
$16: ----
$17: ----
$18: ----
$19: ----
$20: ----
$21: ----



Only the first answer you give to the oracle will be recorded!

HW2: Where Harry Met Sally



Topics:

- ✓ The puzzle your program must solve
- ✓ Command line arguments
- ✓ File I/O (Load_Mem example)
- ✓ #define and enum

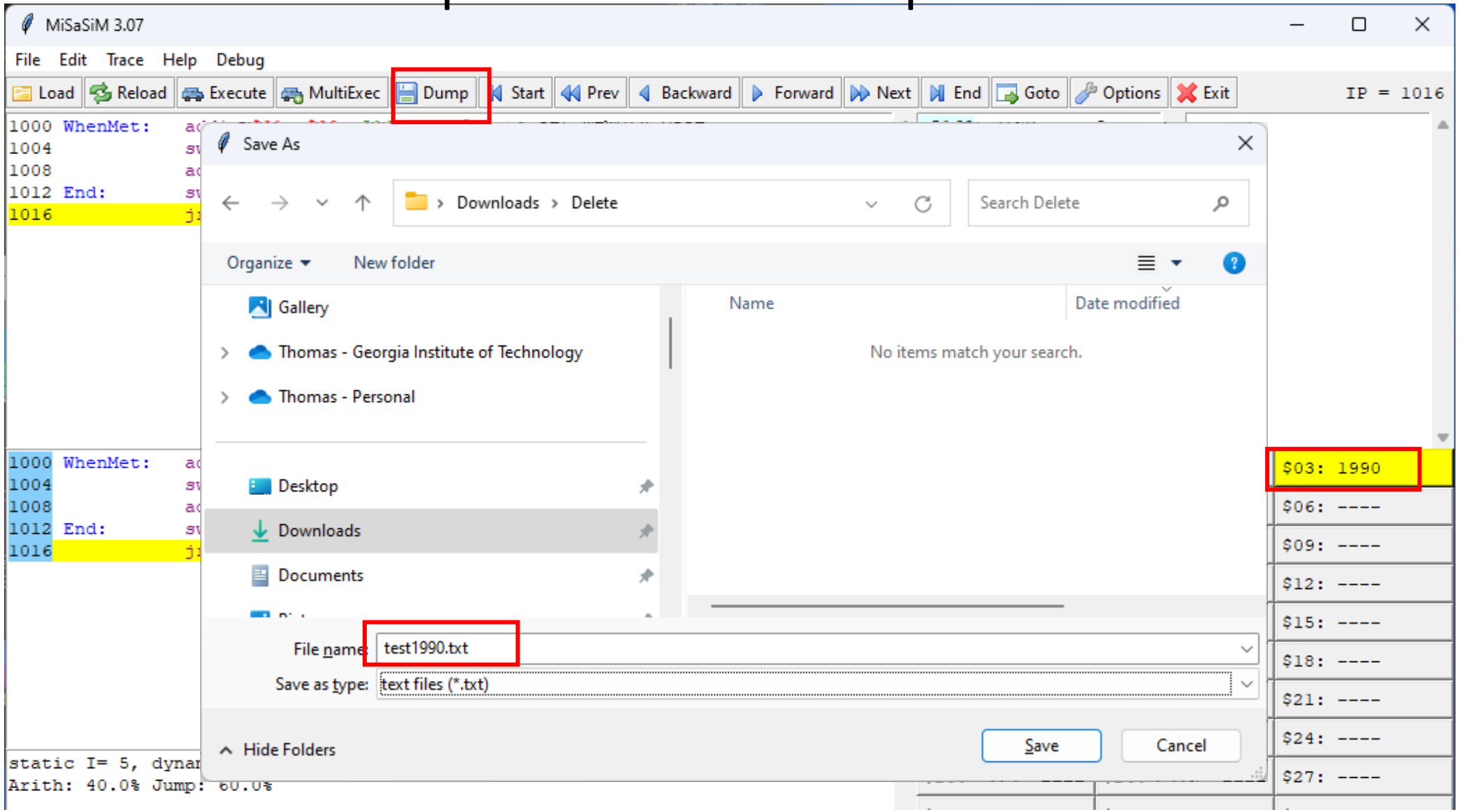
HW2-1.c

- ✓ swi interface for this assignment
- ✓ Oracle

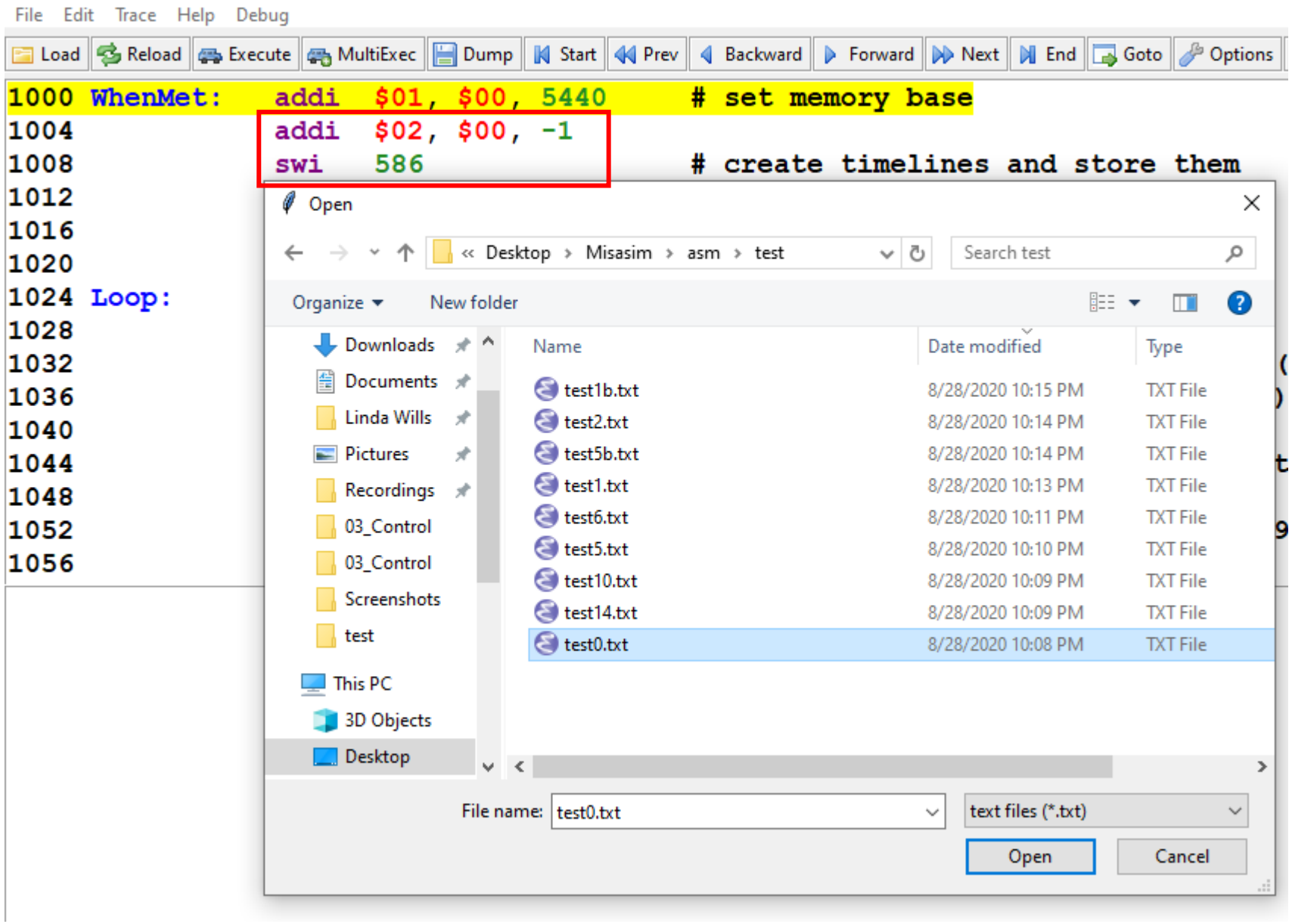
HW2-2.asm

- → Creating test cases

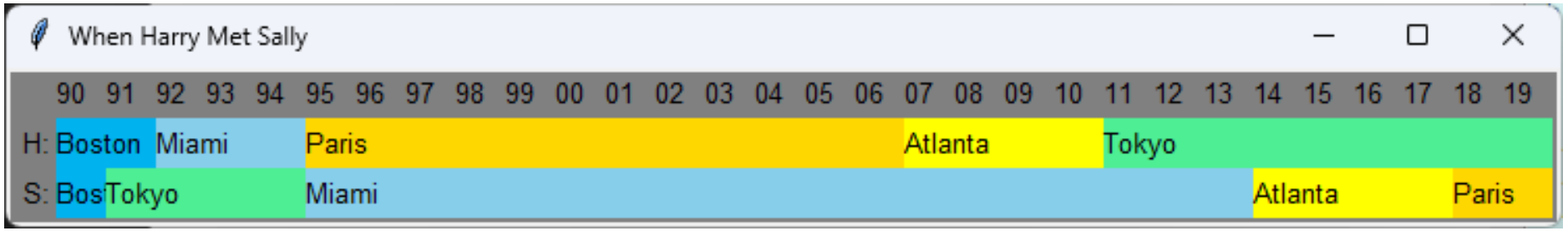
You can dump Misasim output as a new test

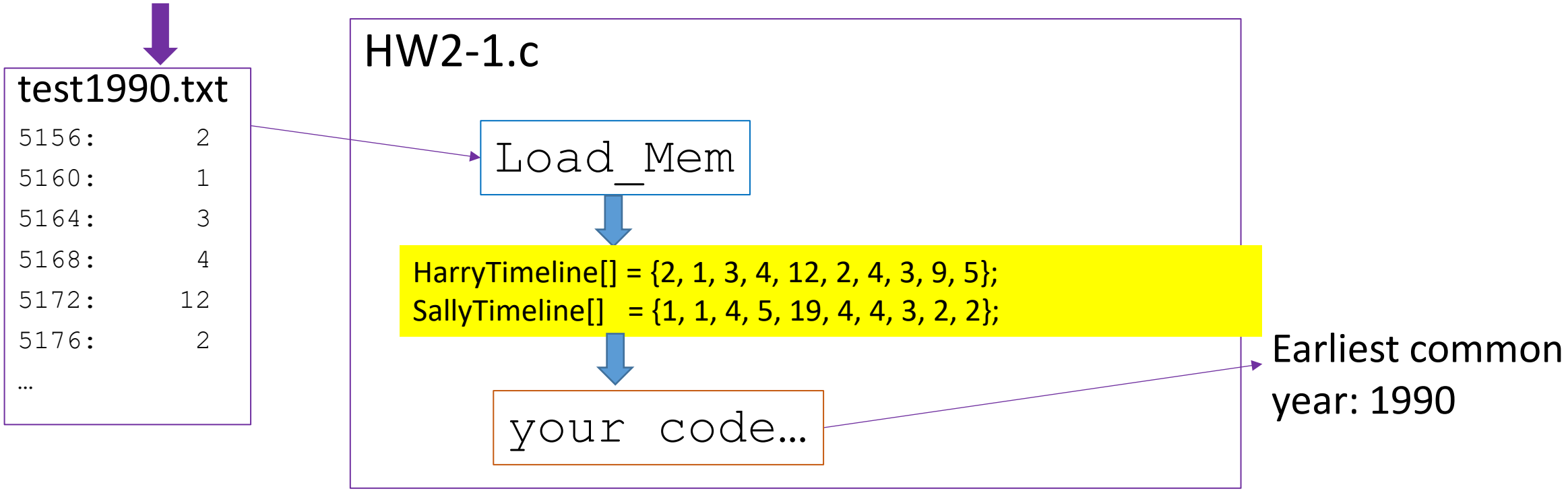


Testcase can be loaded back in… (Easter egg)

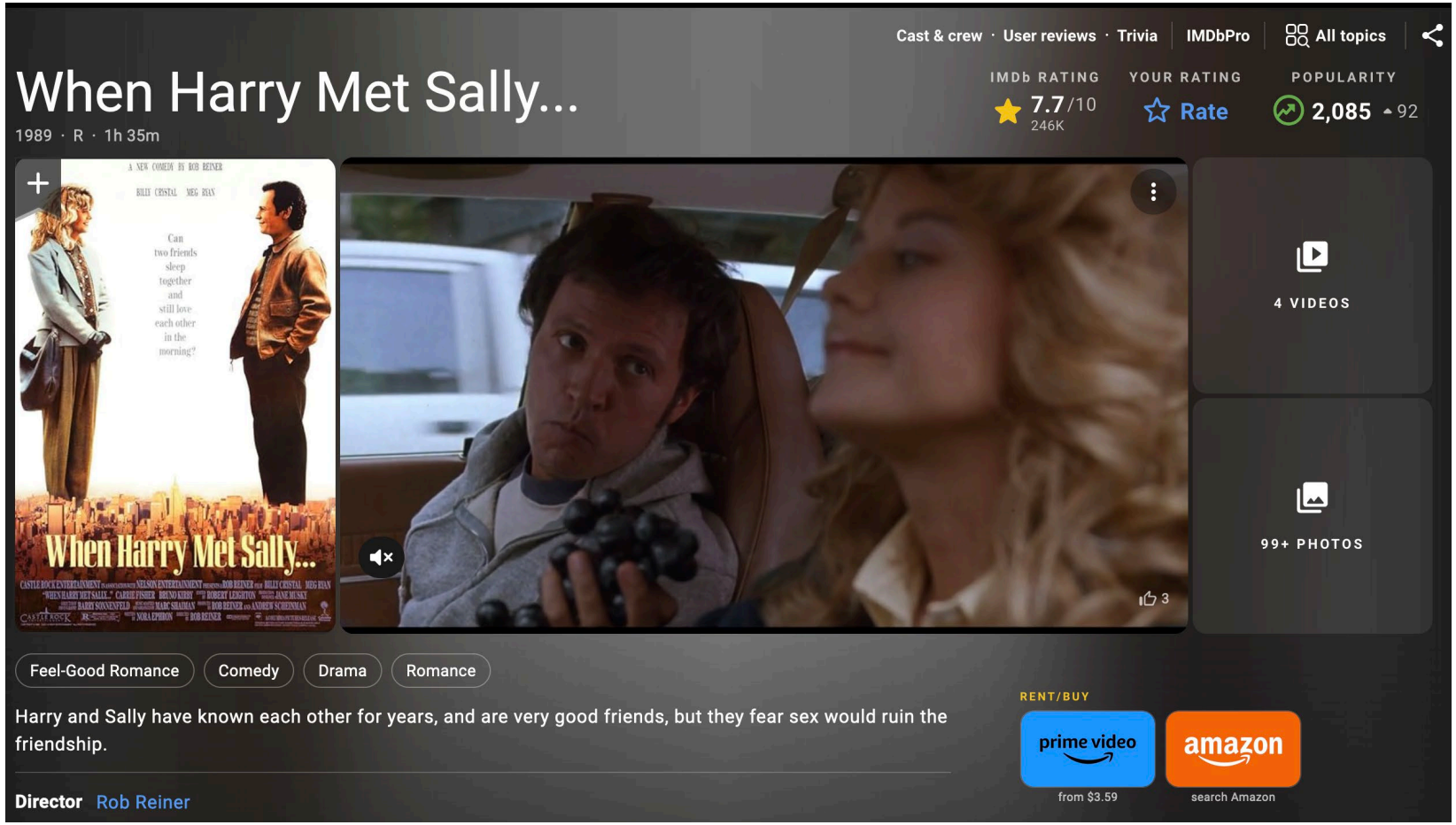


New test cases can also be used for HW2-1.c





When Harry Met Sally (1989)



(Source: IMDB, https://www.imdb.com/title/tt0098635/