Sabancı University
Faculty of Engineering and Natural Sciences
CS305 – Programming Languages
QUIZ II

May 04, 2020

PLEASE NOTE:

- There are 3 questions in this exam.
- Provide only the requested information and nothing more.
- Unreadable, unintelligible and irrelevant answers will not be considered.

Question 1) [**5 points**] Some programming languages require the programmer to release the memory that they dynamically allocate. However, some languages do not have such explicit statements for memory deallocation, but instead this deallocation is performed automatically.

Name the two methods we considered in the class that are used by programming languages to perform these automatic memory deallocations. Also briefly explain these methods in a couple of sentences.

Question 2) [**5 points**] Consider the following C code:

```
main () {
  int A[2][2][3];
  int i,j,k,x;

  x=0;
  for (k=0; k<3; k++) {
    for (j=0; j<2; j++) {
      for (i=0; i<2; i++) {
        x++;
        A[i][j][k] = x;
      }}}
  x=-1;
}
```

when row major is used

A[0, 0, 0] = 1
A[, ,] =
A[, ,] =
A[, ,] =
A[, ,] =
A[, ,] =
A[, ,] =
A[, ,] =
A[, ,] =
A[, ,] =
A[, ,] =
A[1, 1, 2] = 12

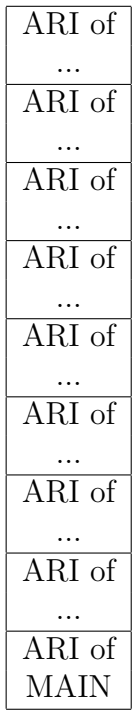
when column major is used

A[0, 0, 0] = 1
A[, ,] =
A[, ,] =
A[, ,] =
A[, ,] =
A[, ,] =
A[, ,] =
A[, ,] =
A[, ,] =
A[, ,] =
A[, ,] =
A[1, 1, 2] = 12

Consider the consecutive blocks of memory given above on the right, where each block can keep an integer value. Assuming that `A[0,0,0]` is stored in the first block, give the elements of the array `A` that will be stored in these consecutive blocks, together with their value, just before the statement `x=-1` is executed.

Question 3) [**5 points**] Consider the following Python program, where we have nested subprogram declarations. `foo_1_1` and `foo_1_2` are declared in `foo_1`. Similarly, `foo_2_1` and `foo_2_2` are declared in `foo_2`. The main part of the program has a call to `foo_2`.

```
def foo_1():
    def foo_1_1():
        x = 1

   def foo_1_2():
       foo_1_1()

   # body of foo_1 starts here
   foo_1_2()

def foo_2():
    def foo_2_1():
        foo_1()

    def foo_2_2():
        foo_2_1()

    # body of foo_1 starts here
    foo_2_2()

# main starts here
foo_2()
```



Consider the moment the program reaches to the statement `x = 1`. Give the activation record instances on the run time stack at that moment. Also give the where the dynamic link and static links of these activation record instances point.

- Use solid lines for dynamic links and dashed lines for static links.
- You don't need to show the other fields of the activation records.
- The number of ARIs given above on the right is not necessarily equal to the number of ARIs that will actually appear on the run time stack when the program reaches to the statement `x = 1`.