March 23, 2015

# CS204 (Adv. Prog.) First Midterm Exam

| 1 | 2 | 3 | 4a | 4b | 4c | 4d | 4e | 5 | 6 | 7 | TOTAL |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | |

**Name and Lastname : İnci Çiftebağlar**
**SUNET ID :**

**Notes: a) Please answer the questions only in the provided space after each question.**
**b) Duration is 100 minutes.**
**c) Closed-book, closed-notes, no calculators and computers. ½ single sided A4 size handwritten cheat-note page is allowed.**
**d) There must be seven pages (including this one) in this booklet. Please check it out!**

## QUESTIONS

**1)** (10 points) What is the output of the following program?

```
#include <iostream>
using namespace std;

int * triple (int num)
{
    num = num*3;
    int * iptr = new int;
    *iptr = num;
    return iptr;
}

int main()
{
    int **ipp = new int *;
    *ipp = new int;
    **ipp = 8;
    *ipp = triple(10);
    cout << **ipp << endl;

    int * ip = triple(11);
    cout << *ip << endl;

    **ipp = *ip;
    if (*ipp == ip)
        cout << "Equal" << endl;
    else
        cout << "Not Equal" << endl;

    **ipp = 3;
    *ip = 4;
    *ipp = ip;
    cout << **ipp << " " << *ip << endl;

    **ipp = 5;
    *ip = 6;
    cout << **ipp << " " << *ip << endl;
    return 0;
}
```

Write your answer here

```
30
33
Not Equal
4 4
6 6
```

NAME:

**2)** (12 points) Consider the given *source.cpp*, *other.cpp*, and *decl.h* files, which are part of the same project in VS2012.

*source.cpp*

```
#include <iostream>
using namespace std;

#define FB "fener"
#define HERYER 61
#define K1905
#define SABRI 51+4
#include "decl.h"
void takil();

int main()
{
   #ifdef K1905
   cout << whereAmI << " " << HERYER+1905;
   cout << " FB:" << FB << endl;
   #endif
   #if HERYER - SABRI == 14
   cout << "Lamborghini" << endl;
   #elif HERYER - SABRI == 6
   cout << "Alfa Romeo" << endl;
   #else
   cout << "Citroen" << endl;
   #endif
   whereAmI = whereAmI + 10;
   takil();
   cout << "main 1: " << whereAmI << endl;
   takil();
   cout << "main 2: " << whereAmI << endl;
   return 0;
}
```

*other.cpp*

```
#include <iostream>
using namespace std;

void takil()
{
   #include "decl.h"
   whereAmI = whereAmI + 1;
   cout << "takil: " << whereAmI << endl;
}
```

*decl.h*

```
static int whereAmI = 13;
```

**a)** What is the <u>translation unit</u> that corresponds to *source.cpp*? Fill in the box below.

```
include <iostream>
using namespace std;
```

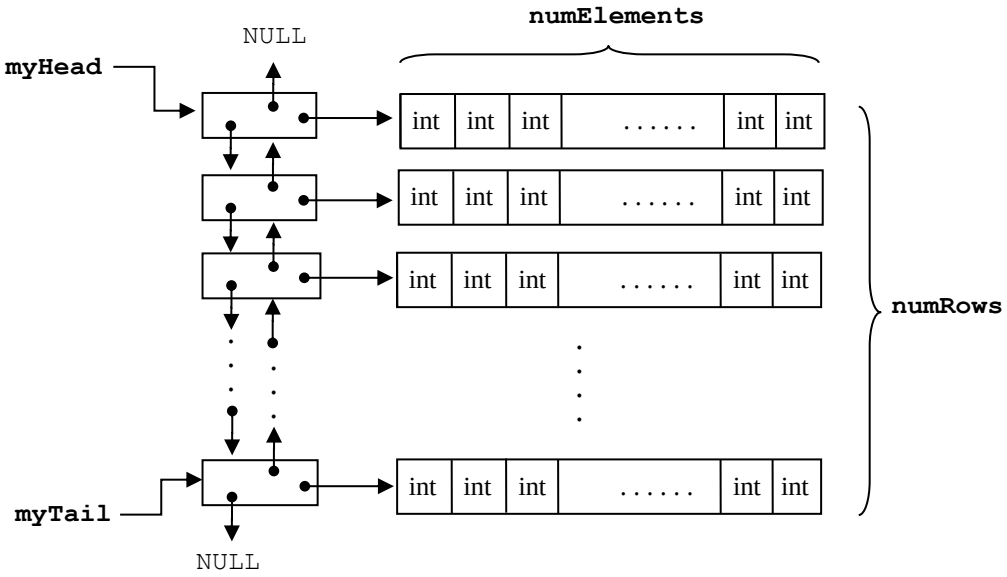
```
static int whereAmI = 13;
void takil();

int main()
{
   cout << whereAmI << " " << 61+1905;
   cout << " FB:" << "fener" << endl;
   cout << "Lamborghini" << endl;
   whereAmI = whereAmI + 10;
   takil();
   cout << "main 1: " << whereAmI << endl;
   takil();
   cout << "main 2: " << whereAmI << endl;
   return 0;
}
```

**b)** What is the output? Give the answer in the box below.

```
13 1966 FB:fener
Lamborghini
takil: 14
main 1: 23
takil: 15
main 2: 23
```

**3)** (18 points) Consider the following dynamic 2D storage class that contains dynamically allocated `numElements` integers in each row. Each row is pointed by a pointer which is part of a doubly linked list node. One of the other fields of the doubly linked list node points to the node for the next row; and another field points to the node of the previous row. Previous of first row node points to NULL and next of last row node points to NULL, as shown in the figure below. `myHead` points to the head of this vertical doubly linked list and `myTail` points to tail of the list.




**a)** Write the node struct declaration for the doubly linked list node. **Hint:** there are only three pointers in this struct.

```
struct node
{
      int * data;
      node * up;
      node * down;
};
```

**b)** Assume that the entire 2D storage is designed as a class called `dynTwoDStorage`. `myHead` and `myTail` are private data members. Write the implementation of member function, called `generateStorage`, that takes `numElements` and `numRows` as two integer parameters, and generates the abovementioned 2D storage. Assume that both parameters are positive.

In this question (both part a and b), you may prefer not to attempt to solve it by signing the "not attempted" box below and secure 4 points. If you sign the "not attempted" box below, you accept that you did not answer this question and you will receive 4 points. In this case, your answer will not be graded even if you write something as solution.

Not attempted

```
void dynTwoDStorage::generateStorage (int numRows, int numElements)
{
      myHead = new node;
      myHead -> up = NULL;
      myHead -> data = new int [numElements];
      node * ptr = myHead;
      int i;
      for (i=1; i<numRows; i++)
      {
            ptr -> down = new node;
            ptr -> down -> up = ptr;
            ptr -> down -> data = new int [numElements];
            ptr = ptr -> down;
      }
      ptr -> down = NULL;
      myTail = ptr;
}
```

NAME:

**4)**

**a)** (4 points) Consider the following macro definition.

```
#define ADD(A, B, C)    (A + B) + C
```

What is the output of the following program piece? Show your work.

```
cout << 20 / ADD(2, 3, 5) << endl;
cout << ADD(2, 3, 5) * 7 << endl;
```

```
9
40
```

**20 / (2 + 3) + 5 = 20 / 5 + 5 = 4 + 5 = 9**

**(2 + 3) + 5 * 7 = 5 + 35 = 40**

**b)** (2 points) Which data structure works in first-in-last-out manner?

**Stack**

**c)** (3 points) What is the output of the following program piece?

```
double * darr = new double [12];
cout << &darr[11] - &darr[4] << endl;
```

```
7
```

**d)** (3 points) Write the necessary code to dynamically allocate 10 strings using `malloc` and assign it to a string pointer `sptr`.

```
string * sptr = (string *) malloc(10*sizeof(string));
```

**e)** (3 points) Write the necessary piece of code in order to call the function `myFunc();` only in **release** configuration.

```
#ifdef NDEBUG
     myFunc();
#endif
```

NAME:

**5)** (15 points)
Write a function that takes a node pointer, say `thePtr`, as a parameter. `thePtr` points to one of the nodes of a **circular** linked list. The function that you will implement will convert the given circular linked list to normal (one-way) linked list using the same nodes (i.e. do **not** generate new nodes and do **not** make a copy operation). The head of the regular linked list will be `thePtr`. The function heading is given below for your convenience. Write the function body within the box.

You do not need to know the entire `node` struct for this question; you just need to know that the next pointer field is named as `next`.

Linked list is **not** implemented as a class.

You may assume that the list is **not empty**. However, there could be any number of elements in it.

In this question, you may prefer not to attempt to solve it by signing the "not attempted" box below and secure 3 points. If you sign the "not attempted" box below, you accept that you did not answer this question and you will receive 4 points. In this case, your answer will not be graded even if you write something as solution.

| Not attempted |
| --- |
| |

```
void antiCircular (node * thePtr)
{
```

```
node * ptr = thePtr;
while (ptr->next != thePtr)
{
          ptr = ptr->next;
}
ptr->next = NULL;
```

```
}
```

NAME:

**6)** (13 points)
Consider the following 2D static integer array definition with 10 rows and 10 columns.

```
int my2D [10][10];
```

Suppose in the program this 2D array is filled with some values. The following piece of code aims to find and display the sum of the 6th row, and the sum of the 8th column, separately.

Complete this piece of code by filling in the boxes with appropriate expressions or **symbols**. You are not allowed to delete or update anything. Moreover, you cannot add anything other than the code that you are going to write in the boxes.

Hint: The processing is done via pointer arithmetic and you have to think in that way.

```
int * ptr;
int sumRow = 0, sumCol = 0;
int k;

ptr = [ my2D[6] ] ;

for (k=0; k<10; k++)
{
      sumRow = sumRow + [ * ] ptr [ ++ ] ;
}

for (k=0; k<10; k++)
{
      ptr = [ my2D[k] ] ;

      sumCol = sumCol + [ * ( ] ptr [ +8) ] ;
}

cout << "Sum of 6th row is " << sumRow << endl;
cout << "Sum of 8th column is " << sumCol << endl;
```

In this question, you may prefer not to attempt to solve it by signing the "not attempted" box below and secure 3 points. If you sign the "not attempted" box below, you accept that you did not answer this question and you will receive 3 points. In this case, your answer will not be graded even if you write something as solution.

| Not attempted |
| --- |
| |

NAME:

**7)** (17 points) First some basic math. An integer `div` is said to be a *divisor* of another integer `num`, if `div` divides `num` with no remainder. *Greatest common divisor* (*gcd*) of two integer numbers `num1` and `num2` is the integer that is the greatest among all of the common divisors of `num1` and `num2`.

The function below is a partial solution for the following problem:

> Write a function that takes two integers, say `num1` and `num2`, as parameters and returns the greatest common divisor of these numbers.

However, the function is incomplete. Complete this function by filling in the boxes with appropriate statements or expressions.

You are not allowed to delete or update anything. Moreover, you cannot add anything other than the code that you are going to write in the boxes. Furthermore, you are not allowed to change the `DynIntStack` class. In other words, you can use only `DynIntStack` class' `push`, `pop`, and `isEmpty` member functions.

**Hint: `isEmpty` is not needed.**

```
int gcd (int num1, int num2)
{
      int i, div1, div2;
      DynIntStack stack1, stack2;

      for (i=1; i<=num1; i++)
            if (num1 % i == 0)
                  stack1.  push(i)  ;

      for (i=1; i<=num2; i++)
            if (num2 % i == 0)
                  stack2.  push(i)  ;

      stack1.  pop(div1)  ;

      stack2.  pop(div2)  ;

      while (  div1 != div2  )
      {
            if (div1 > div2)
                  stack1.  pop(div1)  ;
            else
                  stack2.  pop(div2)  ;
      }

      return  div1  ;   //div2 is also OK
}
```

In this question, you may prefer not to attempt to solve it by signing the "not attempted" box below and secure 4 points. If you sign the "not attempted" box below, you accept that you did not answer this question and you will receive 4 points. In this case, your answer will not be graded even if you write something as solution.

Not attempted