March 28, 2012

CS204 (Adv. Prog.) First Midterm Exam

1	2	3	4	5	6	7	TOTAL

Name and Lastname :
SUNET ID :

Notes: a) Please answer the questions only in the provided space after each question.
b) Duration is 90 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) (9 points) What is the output of the following program?

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

int * f (int * a, int * b)
{
   cout << "f begin: " << *a << " " << *b << endl;
   int * t = b;
   b = a;
   *a = 11;
   cout << "f end: " << *a << " " << *b << " " << *t << endl;
   return t;
}


int main()
{
   int * p = new int;
   int * q = new int;
   *p = 5;
   *q = 1;
    p = f(p,q);
    cout << "main: " << *p << " " << *q << endl;
    return 0;
}
```

NAME:

2) (13 points) Consider the following program and *alex.cpp* file.

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

int main()
{
  #define HAGI 10
  int a = 1;
  #include "alex.cpp"
  cout << a << endl;
  #ifndef _ALEX
  cout << "Bir Alex degil" << endl;
  #endif
  return 0;
}
```

alex.cpp

```
#ifdef HAGI
a = 12;
#endif
#define _ALEX
#define SERCAN  3+1
if (SERCAN * 2 > 6)
cout << "Bir Baros degil" << endl;
cout << HAGI << endl;
```

a) What is the translation unit that corresponds to the main function? Fill in the box below.

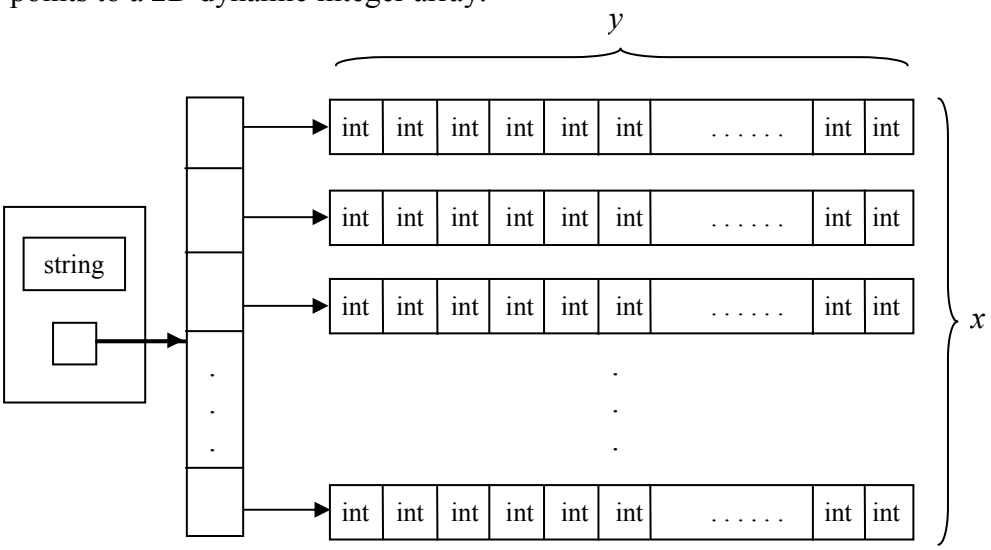
```
#include <iostream>
using namespace std;

int main()
{
```

}

b) Does this program compile and link correctly? If not, specify the erroneous lines in the translation unit. If so, what is the output?

NAME:

3) (18 points) Consider the data structure in the figure below, in which the field of the node points to a 2D dynamic integer array.



a) Write the node struct definition for the node at left hand side of the figure.

b) Write a constructor for that struct that takes *x* and *y* as two integer parameters. The constructor should create the 2D dynamic integer array accordingly (*x* rows and *y* columns) and connect to the struct properly. The string field of struct should be initialized to "CS204" in this constructor.

In this question (in 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

NAME:

4)

a) (2 points) Which of the following identifier names would be the best match for a constant according to our adopted naming style? Pick only one.

```
trabzonfenerneolur

trabzon_fener_ne_olur

Trabzon_Fener_Ne_Olur

trabzon_Fener_Ne_Olur

TRABZON_FENER_NE_OLUR
```

b) (3 points) Rewrite the following declaration using `calloc` statement for dynamic memory allocation?

```
string * sarr = new string [1905];
```

c) (5 points) Consider the following assert statement (assume `x` is an integer variable).

```
assert (x < 13);
```

(i) For which values of `x` does this statement cause the program to abort in debug mode? Explain your reasoning.

(ii) For which values of x does this statement cause the program to abort in release mode? Explain your reasoning.

NAME:

5) (22 points) Consider the following doubly linked list `node` struct definition.

```
struct node {
     int info;
     node *next;
     node *prev;
};
```

Although the node struct is for doubly linked list, it is possible to use only `next` pointers to generate a normal linked list using this structure. Write a function that takes the head pointer of such a normal linked list (not doubly linked) as parameter and converts it to doubly linked list by appropriately connecting the `prev` pointers. The function should also return the rear pointer of the resulting doubly linked list as the function's return value.

You may assume that the list has at least one element. That is, the list is not empty.

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

In this question, you may prefer not to attempt to solve it by signing the "not attempted" box below and secure 5 points. If you sign the "not attempted" box below, you accept that you did not answer this question and you will receive 5 points. In this case, your answer will not be graded even if you write something as solution.

Not attempted

NAME:

6)

a) (6 points) Consider the following `Node` struct definition and declaration of two auto `Node` variables `gs` and `fb`.

```
struct Node {
     int data;
     Node *next;
};

Node gs, fb;
```

Write the piece of code in order to make a circular linked list out of Nodes `gs` and `fb`. Please notice that these two nodes are **not** dynamically allocated.

b) (6 points) What is the output of the following program piece?

```
int i;
int * a = new int [6];
for (i=0; i<6; i++)
  a[i] = i;
int * p = a+1;
for (i=0; i<5; i++)
  cout << *(p+i) + *(a+i) << endl;
```

NAME:

7) (16 points) The function below is a partial solution for the following problem:

> Write a free function (**not** a member function) that takes a `DynIntQueue` as a reference parameter and reverses the order of the entries in this queue. In the implementation, you have to use the help of a stack.

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

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 `DynIntQueue` and `DynIntStack` classes. In other words, you can use only `DynIntQueue` class' `enqueue`, `dequeue`, and `isEmpty` member functions and `DynIntStack` class' `push`, `pop`, and `isEmpty` member functions.

```
void ReverseQueue (DynIntQueue & que)
{
      int temp;
      DynIntStack helper;
      while (!que.isEmpty())
      {
            [                                                    ];
            [                                                    ];
      }

      while (!helper.isEmpty())
      {
            [                                                    ];
            [                                                    ];
      }
}
```