March 25, 2017

CS204 (Adv. Prog.) First Midterm Exam

1	2	3	4a,b	4c.d,e	5	6a	6b	7	TOTAL

Name and Lastname :

SUNET User Name :

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 (half) A4 size, single-sided, 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?

```
int * func (int * p, int * q)
{
  int * temp = new int;
  *temp = 6;
  *p = *p + 1;
  q = p;
  return temp;
}

int main ()
{
  int num = 9;
  int * nump = &num;
  int * mp = new int;
  *mp = 4;
  cout << num << " " << *nump << " " << *mp << endl;
  int * fp = new int;
  *fp = 14;
  fp = func (nump, mp);
  cout << num << " " << *nump << " " << *mp << " " << *fp << endl;

  return 0;
}
```

Write your answer here

NAME:

2) (9 points) What is the translation unit that corresponds to the following program. Give your answer by filling the box on the right.

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

#define F(N) 2*N+1
#define QUIT 27
#define BYE 72

int main ()
{
  #if ( (F(1)) == QUIT/3 )
    cout << "QUIT" << endl;
  #else
    cout << "BYE" << endl;
  #endif
  #ifdef QUIT
    #define BYE
    #define QUIT 32
  #endif
  #ifndef BYE
    cout << QUIT BYE << endl;
  #endif
  #undef BYE
  #ifdef QUIT
  #ifndef BYE
    cout << QUIT+QUIT << endl;
  #endif
    cout << "end" << endl;
  #endif
  return 0;
}
```

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

In this question, you may prefer not to attempt to solve it by signing the "not attempted" box below and secure 2 points. If you sign the "not attempted" box below, you accept that you did not answer this question and you will receive 2 points. In this case, your answer will not be graded even if you write something as solution.

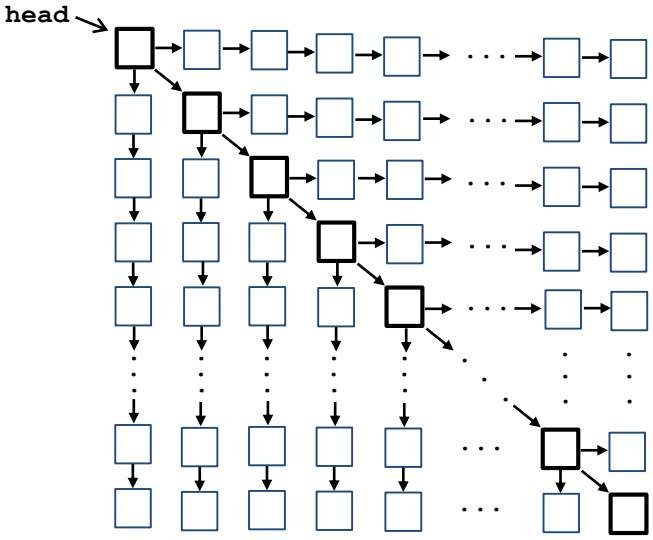
Not attempted

NAME:

3) (18 points) Consider the following dynamic 2D storage. This is a square storage such that the number of rows and columns are the same. There are two different node types (diagonal nodes and non-diagonal nodes). Node structures are given below. The `head` pointer points to element at (0,0) position. In order to reach the other elements, first you have to move through the diagonal and then you have to go to either right or down depending on the indices that you want to reach.

```
struct Node
{
        int info;
        Node *link;
};
```

```
struct DiaNode
{
        int info;
        DiaNode *diagonal;
        Node * right;
        Node * down;
};
```



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

Write a function that takes `head, i` and `j` as parameters and returns the value stored in the node at (`i`, `j`) position, i.e. i^{th} row and j^{th} column, of this 2D storage. The header of the function is given below for your convenience. Please fill the body. Assume that `head` is not NULL; there exists an element at (`i`,`j`) position. Indexing starts from 0. Endpoints of the nodes are all NULL, but this is not important for the solution since we assume that the requested element exists in the storage.

```
int element (DiaNode * head, int i, int j)
{

}
```

NAME:

4)

a) (5 points) Consider the following macro definition.

```
#define SAME(NUM) NUM+NUM-NUM
```

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

```
cout << SAME(7)*SAME(7) << endl;
cout << SAME(7)*(SAME(7)) << endl;
cout << (SAME(7))*SAME(7) << endl;
```

b) (2 points) Write the necessary piece of code in order to display the value of a string variable `myName` only in **debug** configuration. You do not need to define `myName`.

c) (3 points) What is the output of the following piece of program. Show your work.

```
int * p = new int;
*p = 4;
*p = (*p)-- - *p;
cout << *p << endl;
```

d) (3 points) Write a `malloc` statement in order to dynamically allocate 250 `double`s and assign it to a `double` type pointer `dptr`.

e) (5 points) What is the output of the following piece of program.

```
int * p1 = new int;
*p1 = 9;
int *p2 = new int;
*p2 = 11;
int * r;
r = p1 = p2;
cout << *r << " " << *p1 << " " << *p2 << endl;
```

NAME:

5) (15 points) Consider a **doubly linked list** of which the node structure is given below.

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

The following incomplete function aims to return the median value of a doubly linked list given that the list is sorted. The median of a sorted list is defined as the value stored in the middle element or the arithmetic mean of the two middle elements, depending on whether there are odd or even number of elements.

Complete this function by filling in the boxes with appropriate code. 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

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.

```
double median (node * head, node * tail)
{
  node * hp = head;
  node * tp = tail;

  while (                                  &&
                                           )
{

      hp =                                 ;

      tp =                                 ;

}

  return ((double) (hp->info + tp->info))/2;
}
```

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:

6)

a) (9 points) Consider the following piece of program and the corresponding output on the right. Fill in the missing boxes in the program so that it gives the output on the right. 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.

```
0 1 2
10 11 12
20 21 22
30 31 32
40 41 42
```

```
int **    nums;
int  i,j;

nums = [                              ];

for (i = 0; i<5; i++)

      nums[i] = [                              ];

for (i = 0; i < 5; i++)
{
      for (j= 0 ; j< 3 ; j++)
      {
            nums[i][j] = i*10 + j;
      }
}

for (i = 0; i < 5; i++)
{
      [                                                    ]

      for (j= 0 ; j< 3 ; j++)
            cout << *p++ << " ";
      cout << endl;
}
cout << endl;
```

b) (6 points) Consider the given *party.cpp* and *time.cpp* files, which are **part of the same project** in VS2012. What is the output when this project is compiled, linked and executed?

party.cpp

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

static string s1 = "CS204";
string s2 = "Exam";
void Swap ();

int main ()
{
   cout << s1 << " " << s2 << endl;
   Swap ();
   cout << s1 << " " << s2 << endl;
   return 0;
}
```

time.cpp

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

string s1 = "Easy";
extern string s2;

void Swap ()
{
   string temp;
   temp = s1;
   s1 = s2;
   s2 = temp;
   cout << s1 << " " << s2 << endl;
}
```

NAME:

7) (15 points) The factorial of a non-negative integer number x is defined as $x.(x-1).(x-2).(x-3)\cdots 4.3.2.1.$ The factorial of 0 is defined as 1.

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

> Write a function that takes a non-negative integer, say n, as parameter and displays factorials of all numbers between and including 0 and n in reverse order. For example, if parameter n is 7, the function should display `5040 720 120 24 6 2 1 1` as output.

However, the function is incomplete. Complete this function by filling in 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 `DynIntStack` class. In other words, you can use only `DynIntStack` class' `push`, `pop`, and `isEmpty` member functions. Remember that `push` and `pop` are void functions and they take integer parameters. `isEmpty` returns Boolean.

```
void fact_N (int n)
{
      DynIntStack stack;
      int temp;

      stack.                              ;

      for (int i=1; i <= n; i++)
      {
            stack.                        ;

            stack.                        ;

            stack.                        ;

      }

      while (                             )
      {
            stack.                        ;

            cout << temp << " ";
      }
      cout << 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