CS 240 Data Structure

Spring 2018 Exam I

03/01/2018

This exam contains **three section**

A) Code: (basic data type, pointer, ADT)
   a. Reading: Trace the code to predict the output of the code
   b. Filling: Fill in the missing code to guarantee the shown output
   c. Quick short answer telling the difference of the implementation
   d. If the program won't compile, simply say so.

B) Algorithm: (critical thinking and analysis ability)
   a. You are given a sorting problem and you are supposed to design strategies to solve this problem
   b. Briefly talk about how efficient it is. For instance, if the major computation is 2 for loops and each loop contains N iterations, then the complexity is approximately N^2.
   c. You are provided enough space to write out your solution.

C) Implementation: (application of ideas along with learned skills)
   a. Based on your solutions in B, please write up your code. Your code will be judged by the correctness of your interpretation of your solution, not the correctness of your solution because you might have come up with a wrong solution (but that solution be penalized in section B already)
   b. TA will code your program and run it. Based on the correctness of the code (faithful translation of solution and correct execution of the code).

D) Concept
   a. Breaking a long cpp file into main cpp, header file and implementation cpp.

⇨ Good luck and do your best.

⇨ When you are lost in pointers and addresses, the best way to attack that is simply drawing the diagram with respect to address of the variable that the pointer points to, and the pointer's its own address and the dereferencing operator *.

Name:

Score: /80 + 20

A) Basic Data Type, Pointers and ADT: Problem 1: 10pts

```
#include <iostream>

using namespace std;

void changethis(int* p)

{ for(int i = 0; i < 15; i++)

   {  What to fill in here? Answer 1:_________________________________________(5pts)

p++;   }

}

int main ()

{

  int numbers[5];   int * p;

  p = numbers;  *p = 10;  p++;  *p = 20;  p = &numbers[2];  *p = 30;  p = numbers + 3;  *p = 40;

  p = numbers;  *(p+4) = 50; p++ ;

cout<<*(p+4)<<endl;  // What will be printed out here ?? Answer 2: _________________________(5pts)

 int s[15];   int *t;   t = s;

  changethis(t);

   for(int i = 0; i < 15; i++)

    { cout<<s[i]<<endl; } //Here we prints out 1, 3, 5, 7, …, 29, What should you do in Answer 1

  return 0;

}
```

A) Basic Data Type, Pointers and ADT: Problem 2: 10 points

```
#include <iostream>

#include <string>

using namespace std;

void prntarray(int *k, int size);

int main()

{    int thesize = 10;

    int myarray[thesize];

    int *p = myarray;

     for(int i = 0; i<thesize; i++)

    {       myarray[i] = i;     }

prntarray(p, thesize);

 return 0;

}
```

Please write out the function prnarray that will print out the elements inside the myarray array

```
Void prnarray(int* p, int t)

{

Line 1:

Line 2:

Line 3:

Line 4:

Line 5:

Line 6:

Line 7:

}
```

A) Basic Data Type, Pointers and ADT: Problem 3: Pass by reference, pointer and variables

```
// passing parameters by reference
#include <iostream>
using namespace std;
void duplicate (int& a, int& b, int& c)
{  a=2;  b=2;  c=2;}
void duplicate1 (int a, int b, int c)
{  a=5;  b=6;  c=7;}
void duplicate2 (int* a, int* b, int* c)
{  *a = 3;   *b = 3;   *c = 3; }
int main ()
{  int x=1, y=3, z=7;  int *d, *e, *f;
  d=&x;   e=&y;   f=&z;
  duplicate (x, y, z);
// Please predict the output
  cout << "x=" << x << ", y=" << y << ", z=" << z << endl; Answer 3: ______________________ 5pts
  duplicate1 (x, y, z);
  cout << "x=" << x << ", y=" << y << ", z=" << z<<endl; Answer 4: ______________________ 5pts
  duplicate2 (d,e,f );
  cout << "x=" << x << ", y=" << y << ", z=" << z<<endl; Answer 5: ______________________ 5pts
  return 0;
}
```

A) Basic Data Type, Pointers and ADT: Problem 4: Address of an array,

```
int myarray[thesize];
int *p = myarray;
```

Answer 6: True / False: &p = &myarray[0] and myarray = &myarray[0] **(5pts)**

B) Algorithm: Problem 1: Sorting (10 pts + 10 bonus pts)

In PS1, we talked about LinkedList and we tried to sort a given array of integer. In that exercise, we are given an array as the following:

```
int test[] ={4, 8, 2, 1, 5, 6, 66, -11, 34, 7, 3, 21, 17, 7};
```

Please describe how you would sort this array of integers

Answer 6: (10pts)

Step 1: Your idea

Step 2: What is the complexity of your solution and why?

Bonus Answer 6: (5 pts)

Bonus (5 pts): If the given array is only given by the users at runtime (i.e., you do not know the size during the compiling time), how would you handle this problem since now it is not a fixed size 14? (various solutions here, they can range from correct data structure to some size computation function)

Answer 8:

C) Implementation:

Please write out the complete implementation of your solution from section B.

Answer 9: 20pts

```
#include <iostream>

using namespace std;
```

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

11.

12

13.

14.

15.

16.

17.

18.

19

20.

21.

22.

23.

24.

D) **(10pts)** We learned about how to break a long cpp file into multiple files. By doing so, it would be easier to read and debug. Given the old_main.cpp example below, please fill up the new_main.cpp, linkedlist.h and linkedlist.cpp via using those blocks and maybe additional couple lines

Block A

```
#include <iostream>

#include <stdio.h>      /* printf, scanf, NULL */

using namespace std;
```

Block B

```
struct node

{    int value;    node *nxt;};

node* Insert(node *p, int k);

void printLinkedList(node *p);

void addNode(node *p, int k);
```

Block C

```
int main()

{Some operations; Calling node objects;

Operating on node objects and call Insert, printLinkedList functions. }
```

BLOCK D

```
node* Insert(node *p, int k)

{ some implementation; }
```

Block E

```
void printLinkedList(node *p)

{ some implementation; }
```

new_main.cpp

linkedlist.h

linkedlist.cpp

Bonus: Write an one line code to convert military time into standard time. (10pts)

Eg. Military time 1523 will become 3:23 PM; military time 1604 will become 4:04 PM.

Answer: