Section 1 – Multiple Choice – (1 mark each)

1. What is the output of the following program:

```
void fun(int n){
    if (n <= 0) return;
    cout << n + 1;
    fun (n-2);
    cout << n-1;
}
```

```
int main()
{
     fun (3);
     return 0;
}
```

a. 4020
b. 4010
c. 4001
d. 4102
e. 4110
f. None of the above

2. Consider the statement: `char x[8] = "ABC", y[] = "ABC";`
Which of the following is true in C/C++?
a. `x == y`
b. `strcmp(x, y) != 0`
c. `sizeof(y) == sizeof(x)`
d. `strlen(y) < strlen(x)`
e. None of the above

3. Consider the following small program?

```
1 static char s[4] = {0, 3, 4};
2 int main(){
3       s[0] = 'y';
4       s[0] = s[1] + s[2] + s[3];
6       return 0;
7 }
```

a. Program is erroneous because of line 1
b. Program is erroneous because of line 3;
c. Program is erroneous because of line 4;
d. both (b), and (c) are correct statements
e. None of the above

4. Which one of the following statements is **true**?
a. A class constructor can be overloaded
b. A class constructor's return type is NOT void
c. **All of the above is true**
d. None of the above is true

5. Which one of the following statements is **false**?
a. A class destructor can be overloaded
b. A class constructor's return type is NOT `void`
c. Statements (a) and (b) are both false
d. None of the above statements is false

6. Which line in the following code segment may create a compilation error?

```
char m1[] = "\0\0\0";
char *const m2 = &m1[0];
*m2 = 'X';
m2++;
```

a. First line
b. Second line
c. Third line
d. Fourth line
e. None of the lines

7. Which one of the following statements is true:
a. An argument in a C++ function can have a default value only if: All arguments to the right of that argument have a default value.

b. An argument in a C++ function can have a default value only if: All arguments to the left of that argument have a default value.
c. All of the above are correct.
d. None of the above is correct.

8. Which one of the following statements is true:
a. A function in C++ can appear on the left-hand-side of an assignment operator, if its return type is `void`.
b. A function in C++ can appear on the left-hand-side of an assignment operator, if it doesn't have a return type.
c. A function in C++ can appear on the left-hand-side of an assignment operator, if its return type is a reference.
d. There is no way to call a function on the left-hand of an assignment operator.

9. Consider the following program:

```
class Box {
public:
    Box(double x ) { this -> x = x;}
    Box (): x(7777) {}
    double getx() {return x;}
private:
    double x, y;
};
```

```
1   int main(){
2       Box a(7777);
3       Box b = 7777;
4       Box c;
5       Box* d;
6       Box& e = a;
7       Box*& f= d;
8       // Assume more code here
9       return 0;
10  }
```

Which one of the statements in the given main function gives a compilation error?
a. Line 3 in the main
b. Line 4 in the main
c. Line 7 in the main
d. All of the above
e. None of the above

10. Consider the following C++ program and select the best answer:

```
int x;
int main() {
   for(int j = 0; j < 3; j++) fun();
   return 0;
}
```

```
void fun() {
   static int s = 5;
   int m = 2;
   s++;
   x++;
   m++;
   cout << x << s << m;
}
```

a. The output for some data will be garbage.
b. The output of the program is: `163273383`
c. The output of the program is: `163163163`
d. The output of the program is: `163263363`
e. None of the above is true

11. Consider the following code segment, and select the best answer

```
vector <int> v1 = {11, 23, 99};
vector <int> v2 = {3, 9};
v2 = v1;
for(int i =0; i < 3; i++)
    cout << v2.at(i) << " ";
```

The output will be:
a. `11 23 99`
b. `3 23 99`
c. `3 9`
d. `11 23 gargage`
e. There will be no output due to errounous statement(s) in this code segment.

12. Consider the following code segment and select the best answer:

```
string s1 = "863";
s1 += "79";
int y = s1.at(4) - s1.at(2);
```

a. This code gives a compilation error on the second line.
b. The value of y after this code segment will be 6
c. The value of y after this code segment will be -6
d. The value of y after this code segment will be 1

e. None of the above

Consider the following C++ code segment and answer questions 14, 15, 16:

```
1 int main()
2     char *s1;
3     char s2[] = "ABA";
4     strcpy(s1, s2);
5     delete[] s1;
6     return 0;
7 }
```

13. There is a compilation error in this code
    a. true
    b. false

14. There is no logical error on line 4
    a. true
    b. false

15. There is no logical error on line 5
    a. true
    b. flase

Section 2 – Short Answer Questions – 12 marks (Ken)

Part a (4 marks) - A text file named `data.txt` contains following data:

```
      2  527  126
```

What is the output of the following code segment assuming that text file `data.txt` is located in the same working directory and it is successfully accessed.

```
int a[4] = {10, 20, 30, 40};
char filename[100] = "data.txt";
ifstream in(filename);
if(!in){
    printf("File not found.");
    exit(1);
}
in >> a[0] >> a[1] >> a[2] >> a[3];
cout << a[0] << " " << a[1] << " " << a[2] << " " << a[3] << " ";
```

Write your answer in the following space:

2 527 126 40

Part b – Consider the following code segment in **C++**:

```
const char* x[3] = {"012345.987654321", "apple pie", "hello folks"};
ofstream out ("data.bin", ios::binary);
out.write(x[0], sizeof(*x));
cout << out.tellp();
out.close();
```

Now, answer the following questions assuming that operations of opening the file `data.bin` and writing into it have been successful.

Question 1 (1 mark): What would be the size of the file `data.bin` after this code segment?

8 bytes

Question 2 (1 mark): What characters are written into `data.bin`?

012345.9

Question 3 (3 marks): in the following space write a for-loop that uses function `out.write` to write all three elements of array x (including backslash zero) into the binary file that is already open. Hint: you can use any other library function, if needed.

```
for(int i = 0; i < 3; i++)
    out.write(x[i], strlen(x[i]) + 1);
```

Part c (3 marks) – In the space beside the program draw an AR diagram for point one assuming all memory allocations are successful.

```
void fun(int** x,int m){
    *x = new int [m];
}
int main(){
 int *p;
 p = new int[sizeof(int)];
 *p = 222;
 *(p + 1) = 100;
 fun(&p, 4);
 p[1] = 579;
 // point one
 return 0;
}
```

Solution for this AR diagram is not available

Section 3: Class Activation Record and Program Output Determination (19 *marks total*)

Consider the following C++ program and answer the questions a, and b on the next page.

```
char m[] = "Bar";
```

```
class MyString {
    char *text;
    unsigned long length;
public:
    MyString(const char* s);
    MyString();
    ~MyString();
    MyString(const MyString& souce);
    MyString (int n);
    void set_txt(const char* s);
    char* fun(const MyString& txt)const;
};

MyString::MyString(const char* s): length(strlen(s)), text(new char[strlen(s) + 1]){
    for(int i = 0; i <= length ; i++)
        text[i] = s[i];
    cout << "MyString ctor-1"<< endl;
}

MyString::MyString(int n):length(n), text(new char[n + 1]){
    for(int i = 0; i <= length ; i++)
        text[i] = 0;
    cout << "MyString ctor-2" << endl;
}

MyString::MyString():length(0), text(new char[1]){
    text[0] = '\0';
    cout << "MyString Default Ctor"<< endl;
}

MyString::MyString(const MyString& source):length(source.length) {
    cout << "copy ctor" << endl;
    text = new char[length + 1];
    for(int i = 0; i <= length ; i++)
        text[i] = source.text[i];
}

MyString::~MyString() {
    delete[] text;
    cout << "MyString dtor" << endl;
}

void MyString::set_txt(const char* s) {
    int i;
    for(i = 0; i < strlen(s) ; i++)
        text[i] = s[i];
    text[i] = 0;
}

char* MyString::fun(const MyString& txt)const{
    // point one
    return txt.text;
}
```

```
class Magazine {
public:
    Magazine(int i, int size): id(i), doc(size){}
    void fun(const MyString& txt);
    MyString& get() {return doc;}
private:
    int id;
    MyString doc;
};

void Magazine::fun(const MyString& txt){
    cout << txt.fun("TM") << endl;
}
```

```
int main() {
    Magazine* q= new Magazine(100, 4);
    m[0] = 'F';
    q -> get().set_txt(m);
    {
        Magazine b(*q);
        b.get().set_txt("B");
        b.fun(b.get());
    }
    delete q;
    return 0;
}
```

If you want you can detach this page, but you must write your student id on this page and return it back to us at the end of the exam. **Please don't write anything on this page.**

Your Student ID Number:

Question a (12 marks) - Draw a memory diagram for point one in the following space:

Solution for this AR diagram is not available

Question b (7 marks): What is the output of the program? Write your answer in the following space:

<u>Answer is:</u>

```
MyString ctor-2
copy ctor
MyString ctor-1
TM
MyString dtor
MyString dtor
MyString dtor
```

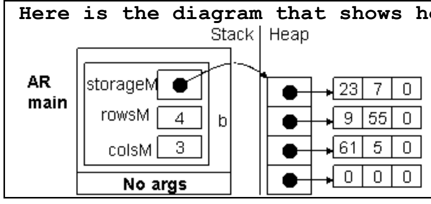
Section 4 (total of 10 marks) – Writing a member function for a class called Matrix (Masoud)

Consider the **partial-definition** of the class **Matrix** in the following box and answer the following question:

```
class Matrix {
public:
    Matrix(int n, int m);
    ~Matrix();
    void set(int row, int col, int value) {storageM[row][col] = value;}
    void resize (int new_row, int new_column);
    /* PROMISES: if new_row or new_column is less than or equal zero does nothing. Otherwise,
     * dynamically reallocates the space for storageM to resize the matrix to new_row by
     * new_colum. If the new_row or new_column is less than this->rowsM or this->colsM the
     * data in resized matrix for the corresponding elements of the original matrix will be
     * lost. If the new_row or new_column is greater than this->rowsM or this->colsM the
     * the extra elements in the resized matrix will be all set to zero. */
private:
    int rowsM;                  // number of rows in a Matrix
    int colsM;                  // number of columns in a Matrix
    int ** storageM;
};
Matrix::Matrix(int r, int c): colsM(c),rowsM(r){
    storageM = new int* [rowsM];
    assert(storageM != nullptr );
    for(int i=0; i<r; i++){
        storageM[i] = new int[colsM];
        assert(storageM[i] != nullptr );
        for(int j=0; j < colsM; j++)
            storageM[i][j] = 0.0;
    }
}
```

```
int main(void) {
    Matrix  b (4, 3);   // creates a Matrix with 4 rows and 3 columns all initialized to zero
    b.set(0, 0, 23);    // store 23 in the first row, first column
    b.set(0, 1, 7);     // store 7 in the first row, second column
    b.set(1, 0, 9);     // store 9 in second row, first column
    b.set(1, 1, 55);    // store 55 in second row, second column
    b.set(2, 0, 61);    // store 61 in third row, first column
    b.set(2, 1, 5);     // store 5 in third row, second column
    // Point one
    return 0;
}
```

Here is the diagram that shows how object b appears on the memory at point one:



Question - Write the definition of member function `resize` in following space.

```
void Matrix::resize(int r, int c){
    if (r <=0 || c <=0) retrun;
    int **p;

    p = new int*[r];
    if(p == nullptr) {
        cerr << " no memorey allocated";
        exit(1);
    }

    for (int i = 0; i < r; i++){
        p[i] = new int [c];
        if(p[i] == nullptr) {
            cerr << " no memorey allocated";
            exit(1);
        }
        for(int j = 0; j < c; j++)
            p[i][j] = 0;
    }

    int rsize, csize;

    if(r > rowsM)
        rsize = rowsM;
    else
        rsize = r;

    if(c > colsM)
        csize = colsM;
    else
        csize = c;

    for(int i =0; i < rsize; i++) {
        for(int j = 0; j < csize; j++)
            p[i][j] = storageM [i][j];
        delete [] storageM[i];
    }

    delete [] storageM;
    storageM = p;
    rowsM = r;
    colsM = c;
}
```

Section 5: (*8 marks*) Writing class constructor (Ken)

Consider the partial definition of `struct String` and class `StringArray` that is used to create objects that contains an array of `struct String`. Then answer the question on the next page.

```
struct String {
  int lengthM;   //the string length - number of characters excluding \0
  char* stringM; //a pointer to be used for dynamic allocation of an array of characters, with lengthM + 1
};
```

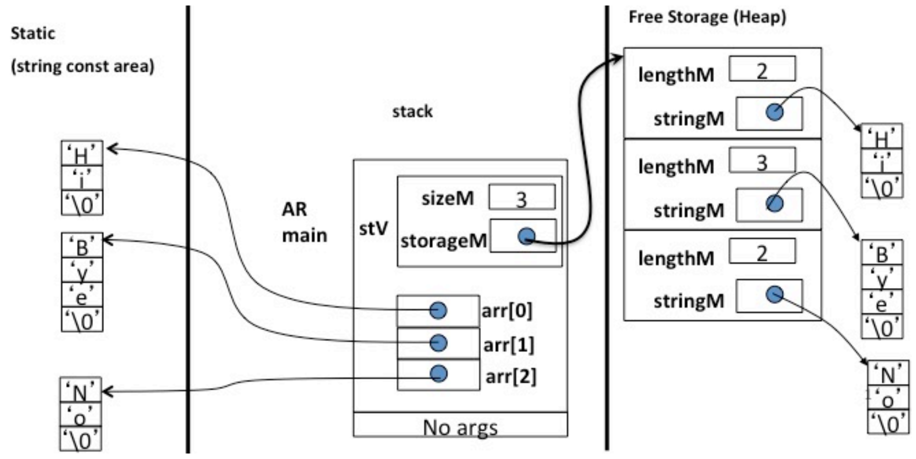
```
class StringArray {
public:

    StringArray(const char* arr[], int n);
    /* REQUIRES: n > 0 and arr points to an array of n elements that each element is a
     *           char pointer pointing to a c-string.
     * PROMISES: creates an array of n String objects that storageM[i].stringM points
     *           to a dynamically allocated memory on the heap, holding the copy of c-
     *           string pointed by arr[i].
     */


    StringArray (const StringArray& src);

    ~StringArray();

    StringArray& operator = (const StringArray& v);

    int size() {return sizeM;}


private:

  String* storageM; // pointer to dynamically allocate space of an array of struct String object

  int sizeM;       // number of elements in the StringArray

};
```

```
int main()
{
  const char* arr[3] = {"Hi", "Bye", ""}; // Note: no space between last two quotation marks

  StringArray ms (arr, 3);

  // Point  one

  return 0;
}
```

If you want you can detach this page, but you must write your student id on this page and return it back to us at the end of the exam. **Please don't write anything on this page.**

Your Student ID Number:

Question – Please read the given function interface comment for the constructor of class StringArray and write its implementation in the following space. To help you to better understand how this constructor should work the AR diagram for point one in the main function is here:



Write the implementation of the constructor of class `StringArray` in this space:

```
StringArray::StringArray(const char* arr[], int n) {
    storageM = new String [n];
    if(storageM == 0 ) {
        cerr << "    ";
        exit(1);
    }

    for (int i = 0; i < n; i++){
        storage[i].lengthM = strlen(arr[i]);
        storageM[i].stringM = new char[storage[i].lengthM + 1];
        if(storageM[i].stringM == 0 ) {
            cerr << "    ";
            exit(1);
        }
        strcpy(storageM[i].stringM, arr[i]);
    }
    sizeM = n;
}
```

Section 6 - Linked List (20 marks)

Consider the **partial** definition of a class called `SimpleList` and a class `Node` and answer the following questions:

```
class Node {
public:
  Node(int aItem): item(aItem), next(0) {}
  int item;
  Node *next;
};

class SimpleList {
public:
 SimpleList(): headM(0) {};
 void insert(const int& itemA);

 void insert_last(const int& itemA);
 /* PROMISE: inserts a node object with the itemA to the end of the list */

 ~SimpleList();
 SimpleList& operator= (const SimpleList& rhs);
private:
  Node *headM;
};
```

```
void SimpleList::insert(const int& itemA){
  Node *new_node = new Node(itemA);

  if (headM == 0 || itemA <= headM->item ) {
    new_node->next = headM;
    headM = new_node;
  }
  else {
    Node *before = headM;
    Node *after = headM->next;

    while(after != 0 && itemA > after->item) {
       before = after;
       after = after->next;
    }
     new_node->next = after;
     before->next = new_node;
      // point ONE
    } // end of else
}
```

```
int main(){
  SimpleList the_list;
  the_list.insert(440);
  the_list.insert(220);
  the_list.insert(330);
  the_list.remove(550);
  return 0;
}
```

Part a (7 marks) Draw a memory diagram for point one, when reached for the **first time**.

Solution for this AR diagram is not available

Part b (7 marks) Write the definition of the **assignment operator** for class SimpleList, in the following space:

SOLUTION

```
SimpleList& SimpleList::operator= (const SimpleList& rhs){
    if(&rhs != this) {
        while(headM) {
            Node *p = headM;
            headM = headM -> next;
            delete p;;
        }

        Node *p = rhs.headM;
        while(p){
            Node *new_node = new Node(p->item);
            if (headM == 0 ) {
                new_node->next = headM;
                headM = new_node;
            }
            else {
                Node *before = headM;
                Node *after = headM->next;

                while(after != 0 ) {
                    before = after;
                    after = after->next;
                }
                before->next = new_node;
            } // end of else
            p = p ->next;
        }
    }
    return *this;
}
```

Part c (4 marks) Write the definition of function `insert_last` that inserts a node to the end of the list:

```
void SimpleList::insert_last(const int& itemA){
    Node *new_node = new Node(itemA);
    if (headM == 0 ) {
        new_node->next = headM;
        headM = new_node;
    }
    else {
        Node *before = headM;
        Node *after = headM->next;

        while(after != 0 ) {
            before = after;
            after = after->next;
        }
        before->next = new_node;
    } // end of else
}
```

Part d (2 marks) in the following space write the definition of the destructor for class SimpleList, using recursive technique. **No marks will be considered for non-recursive solutions:**

The purpose of this question is to test students' ability to write a recursive function. In practice such a recursive solution for removing nodes of a linked list is not good solution. In general, we normally do not call destructors, as they will be called automatically when an object goes out of scope.

```
SimpleList::~SimpleList(){
    if(headM == 0)
        return;
    headM = headM -> next;
    ~SimpleList();
    delete headM;
}
```

Section 7 - Drawing an AR Diagram for a Recursive Call - Mahmood

Part a (7 marks) - Draw a memory diagram when the value of `result` in the following program is `13`.

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


int fun(int *a, int start, int end);

int main(void)
{
   int a[] = {5, 8, 1, 9, 11};

   int result = fun(a, 0, 4);

   cout << "the result is: " << result << endl;

   return 0;
}

//-------------------------

int fun(int *a, int start, int end)
{
   int Rval, Lval, result;

   if(start == end)
      return a[start];

   else
  {
      int m = (start + end)/2;

      Lval = fun(a, start, m);

      Rval = fun(a, m + 1, end);
   }

   return result = Lval + Rval;
}
```

```
// Draw your diagram in this box
```

Solution for this AR diagram is not available

Part b (2 marks) - What is the output of the program?

Part b (6 marks) In this section you should write the recursive solution of a function that returns the value of x^n

```
int power(int x, int n);
/* REQUIRES: n >= 0
 * PROMISES: Returns x to the power of n. Example: when x is 2 n is 3 should return 8. */
```

Here is a sequence of numeric solutions that shows a recursive pattern that you should use in this solution.

$\mathbf{x^{18} = x^{18/2} * x^{18/2} = x^9 * x^9}$

$\mathbf{x^9 = x^{9/2} * x^{9/2} * x = x^4 * x^4 * x}$

$\mathbf{x^4 = x^{4/2} * x^{4/2} = x^4 * x^4}$

$\mathbf{x^2 = x^{2/2} * x^{2/2} = x * x}$

$\mathbf{x^1 = x^0 * x^0 * x}$

```
int power(int x, int n){
    int p;
    if(n == 0)
        p = 1;
    else if(n % 2 == 0)
        p = power(x, n/2) * power(x, n/2);
    else
        p = power(x, n/2) * power(x, n/2) * x;

    return p;
}
```

Section 8 (total of 9 marks) Masoud

In the area beside the following C++, draw a memory diagram for point one:

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

char bar[] = {"ABCCD"};

void fun(char **z, int n){
  *(*(z + 2) - 2) = 'T';
  z[2][-3] = 'P';
  (*z)[1] = 'N';
  *z[1] = 'V';

  // point one
}

int main(void){
  char pc[2] = {'\0'};
  char *y[3];

  char *x = new char[sizeof(x)];
  strcpy(x, "Box");
  y[0] = bar + 2;
  y[1] = pc;
  y[2] = &x[3];
  fun(y, 3);
  return 0;
}
```

Solution for this AR diagram is not available

Section 9 (total of 7 marks)

In the following space write a C++ program that reads a text file with zero or more integer number, and creates an output text file, in which the numbers will be stored in tabular format with three columns and several rows, as needed. Here are a couple of examples of input files and expected output files:

If input file contains	Output file will be (two spaces between numbers in each row)

22   31   55   88   99   10
61   20   12   41
18
	22 31 55
	88 99 10
	61 20 12
	41 18

22 31 55 88	22 31 55 88

Note: The data in the output file can be separated by one or more spaces. It is up to you.

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

int main() {
    ifstream in ("input.data");
    if(!in) {
        cerr << "cannot open the input file.";
        exit(1);
    }
    ofstream out("output.data");
    if(!out) {
        cerr << "cannot open the output file.";
        exit(1);
    }

    int x;
    int col = 0;
    in >> x;

    while(!in.eof()) {  // enter the loop if last read was not end of file
        if(in.fail()){
            cerr << "invalid data in the input file" << endl;
            exit(1);
        }
        if(col < 3) {
            out << x << "  ";
            col++;
        }
        if (col == 3){
            col = 0;
            out << "\n";
        }
        in >> x;
    }
    in.close();
    out.close();
    return 0;
}
```