ARRAYS, POINTERS, PARAMETER PASSING

Question 1 (25 points)

```
#include <iostream.h>
class SA
{ public:
        int *s, a;
        SA(int &o, int aa) {s=&o; a=aa;}
        void A(int &n) {*s=*s**s;  n=a*n  ;sa();}
        void B(int *n) {*s=*s**s; *n=a**n ;sa();}
        void C(int  n) {*s=*s**s;  n=a*n  ;sa();}
        void D(int *n) { s=&a   ; *n=a**n ;sa();}
        void sa() {cout <<"SA:"<<*s<<":"<<a<<endl;};
};
int main(void)
{
        int i=1, j=2, k=3;
        SA x(i,j);
        x.sa();
        x.A(k);
        x.A(x.a);
        x.B(x.s);
        cout<<k<<endl;
        x.C(k);
        x.D(&x.a);
        x.D(x.s);
}
// Write down the output of the program.
```

Given above C++ program; write down the output, indicate the erroneous step(s) if any. Justify your answer.

1: SA:1:2
2: SA:1:2
3: SA:1:4
4: SA:4:4
5: 6
6: SA:16:4
7: SA:16:16
8: SA:256:256
9:
10:

Question 2

Find out the output produced by the following programs

a)

```
#include <iostream>
void main ()
{ int i;
  int numbers[6];
  int * p;
  p = numbers;
  for (i=0; i<6;i++){
 *(p+i)=i;
  }
  p = &numbers[2];  *p = 29;
  p = numbers + 3;  *p = 17;

  p= numbers; *p = 21;
  p++;  *p = 32;
  p = numbers;  *(p+4) = 16;
  for (int n=0; n<6; n++)
    cout <<"Line"<<n<<":"<< numbers[n] <<"\n";
}
```

Output:

```
Line 0: 21
Line 1: 32
Line 2: 29
Line 3: 17
Line 4: 16
Line 5: 5
```

b)

```
#include <iostream>

void  main() {
    const int size=5;
    int B[size][size];

    for (int r=0; r<size; r++) {
        for (int c=0; c<size; c++) {
            B[r][c]=10*r+c;
        }
    }

    for (r=1; r<size; r++) {
        for (int c=1; c<=r; c++) {
            if (r==c) {
               B[r][c]=1;
            } else {
               B[r][c]=0;
            }
        }
    }

    cout<< "Array B is"<<endl;

    for (r=0; r<size; r++) {
        for (int c=0; c<size; c++) {
            cout<<B[r][c]<<" ";
        }
        cout << endl;
    }
    cout << endl;
}
```

Output:

```
Array B is
0 1 2 3 4
10 1 12 13 14
20 0 1 23 24
30 0 0 1 34
40 0 0 0 1
```

Question 3 (25 points)

```
#include <iostream.h>

void triple(double &num);

main()
{
      double d=10.0;
      triple(d);
      cout<<d;
      return 0;
}

//Triple num's value
void triple(double &num)
{
      num=3*num;
}
```

(A)

```
#include <iostream.h>

void triple(double &num);

main()
{
      double d=10.0;
      triple(&d);
      cout<<d;
      return 0;
}

//Triple num's value
void triple(double &num)
{
      num=3*num;
}
```

(B)

```
#include <iostream.h>

void triple(double *num);

main()
{
      double d=10.0;
      triple(&d);
      cout<<d;
      return 0;
}

//Triple num's value
void triple(double *num)
{
      *num=3**num;
}
```

(C)

```
#include <iostream.h>

void triple(double num);

main()
{
      double d=10.0;
      triple(d);
      cout<<d;
      return 0;
}

//Triple num's value
void triple(double num)
{
      num=3*num;
}
```

(D)

(a) Given above C++ programs; write down the output of each, indicate the erroneous program(s) if any. Justify your answers.

Program (A): Output: 30 (parameter is passed by reference)
Program (B): **incorrect !!** (error: `could not convert `&d' to `double&'`)
Program (C): Output: 30 (passed by reference with pointer implementation)
Program (D): Output: 10 (parameter is passed by value)

(b) Given the following C++ program:

```
#include <iostream.h>
void ArrayExchange(int ....a, int ....b)
{
        int k, temp;
        for (k=0;k<4;k++)
```

```
        {
                    temp= ....b[k];
                ....b[k]= ....a[k];
                ....a[k]=temp;
        }
}
int main()
{
        int *pi, *pj, l;
        int i[]={0,1,2,3};
        int j[]={5,6,7,8};
        .... = ....;
        .... = ....;
        cout <<"Original\ti: ";for(l=0;l<4;l++) cout<<i[l]<<'\t';cout<<endl;
        cout <<"Original\tj: ";for(l=0;l<4;l++) cout<<j[l]<<'\t';cout<<endl;
        ArrayExchange(....pi,....pj);
        cout <<"Exchanged:\ti: ";for(l=0;l<4;l++) cout<<i[l]<<'\t';cout<<endl;
        cout <<"Exchanged:\tj: ";for(l=0;l<4;l++) cout<<j[l]<<'\t';cout<<endl;
}
```

Fill in the blanks, i.e., (**. . .**), with appropriate operators if required in order to make the program work as desired. Briefly justify your reasoning.

Solution:

```
#include <iostream.h>
void ArrayExchange(int *a, int *b)
{
        int k, temp;
        for (k=0;k<4;k++)
        {
                temp=b[k];
                b[k]=a[k];
                a[k]=temp;
        }
}
int main()
{
        int *pi, *pj, l;
        int i[]={0,1,2,3};
        int j[]={5,6,7,8};
        pi=i;
        pj=j;
        cout <<"Original:\ti: ";for(l=0;l<4;l++) cout<<i[l]<<'\t';cout<<endl;
        cout <<"Original:\tj: ";for(l=0;l<4;l++) cout<<j[l]<<'\t';cout<<endl;
        ArrayExchange(pi,pj);
        cout <<"Exchanged:\ti: ";for(l=0;l<4;l++) cout<<i[l]<<'\t';cout<<endl;
        cout <<"Exchanged:\tj: ";for(l=0;l<4;l++) cout<<j[l]<<'\t';cout<<endl;
}
```

The output of the program is:

```
Before:   i: 0    1       2       3
Before:   j: 5    6       7       8
After:    i: 5    6       7       8
After:    j: 0    1       2       3
```

(c) Briefly describe two main approaches to provide a function with argument along with their differences if any.

BY VALUE: The calling program copies the actual full object value to the local data area of the called program.

BY REFERENCE: The address of the object as stored by the calling program is passed into the called program which operates on the original data; not its local copy.

Question 4) a) (8 pts) Write the program outputs by tracing the following piece of code:

```
#include<stdio.h>
#include<iostream.h>
void main()
{
        int *a, c, d=5;
        int *b=&d;
        c=2;
        a=&c;
        *a=d+12;
        cout<<"*a:"<<*a<<"*b:"<<*b<<"\n";
        cout<<"c:"<<c<<"d:"<<d<<"\n";
        cout<<"a:"<<a<<"b:"<<b<<"\n";
        a=b;
        *b=*a+5;
        cout<<"*a:"<<*a<<"*b:"<<*b<<"\n";
        cout<<"c:"<<c<<"d:"<<d<<"\n";
        cout<<"a:"<<a<<"b:"<<b<<"\n";
}
```

Outputs:
*a:17*b:5
a:17d:5
a:addres1b:address2
*a:10*b:10
c:17d:10
a:addressb:address

b) (17 pts) Write or draw diagrams to show the values of the variables (a,b,c,d,f,A) at the control point 1 and control point 2.

```
int F(int *x, int *&y,int&z)
{
        int *w=y;
        y=&z;
        *y=*(x+1);
        x[2]=(*w)*z;
        x=x+2;
        return *x;
}
void main(void)
{
        int a=20, b=100, f;
        int A[3];
        int *c=&b;
        int *d=A+1;
        for (int i=0; i<3; i++)
                A[i]=3*(i+1);
//control point 1
        cout<<"a:"<<a <<"b:"<<b<<"\n";
        cout<<"*c:"<<*c<<"*d:"<<*d <<"*A:"<<*A<<"\n";
        cout<<"A[0]:"<<A[0]<< "A[1]:"<<A[1]<< "A[2]:"<<A[2]<<"\n";

        f=F(A,c,a);
//control point 2
        cout<<"a:"<<a <<"b:"<<b<<"\n";
        cout<<"*c:"<<*c<<"*d:"<<*d <<"*A:"<<*A<<"\n";
        cout<<"A[0]:"<<A[0]<< "A[1]:"<<A[1]<< "A[2]:"<<A[2]<<"\n";
        cout<<"f:"<<f<<"\n";
}
```

Outputs:
a:20b:100
*c:100*d:6*A:3
A[0]:3A[1]:6A[2]:9
a:6b:100
*c:6*d:6*A:3
A[0]:3A[1]:6A[2]:600
f:600

Question 5. A) (6 pts) Below are some lines from C++ codes. State if the underlined C++ instructions are correct or not. Explain your reasoning.

a)

```
int a=50;
char x[a];
```
//Correct/Not correct? Why? Not. a is not constant integer!

b)

```
void functionx(int *x)
{
     //do something
}
void main ()
{ int A[]= {0,1,2,3,4,5};
  functionx(A);
 //do something}
```
//Correct/Not correct? Why? Correct. A is also a pointer to an integer.

c)

```
char A[]={'a','b','c'};
A = "Hi";
```
//Correct/Not correct? Why? Not Correct. An array cannot be assigned by a string.

Question 5. B) (7 pts) What will be the output of the following piece of code? When contents of a memory location are not known, indicate this.

```
int i, j, M[3][4];
int temp;
for(i = 2; i >= 0; i = i - 1)
{  for(j = 3; j >= 0; j = j - 1)
     { if (i==j) M[i][j]=10;
             else     M[i][j]=i*j;
             cout<< i<<","<< j<<":"<< M[i][j]<<"; "; }
      cout<<endl; }
temp=*(M+5);
cout<<M[0][0]<<'and'<< temp << endl;
```

2,3:6; 2,2:10; 2,1:2; 2,0:0;
1,3:3; 1,2:2; 1,1:10; 1,0:0;
0,3:0; 0,2:0; 0,1:0, 0,0:10;
10 and 2 //M[1][1] will be printed out, which is 2.

Question 5. C) (6 pts) Class Rectangle is declared and implemented as below.

```
class Rectangle
{ private:
      float length;
      float width;
  public:
      Rectangle(float l, float w)          /* constructor function */
      {     length=l;
            width=w; };
      float Circumference(void) const
      {     return 2*(length+width); };
      float Area(void) const
      {     return (length*width); };
 }
```

What will be the output of the following piece of code? When contents of a memory location are not known, indicate this.

```
float a,b,c;
Rectangle *p;
Rectangle R[3]= { Rectangle (5.0, 7.0), Rectangle (5.0, 6.0), Rectangle(5.0, 6.0)};
p=R++;
a=*p.Area();
b=R[0].Circumference();
c=R[3].Area();
cout<<"a is "<<a<<", "<<"b is "<<b<<", "<<"c is "<<c<<".";
cout<<endl;
```

a is 30, b is 24, c is XX.

Question 5. D) (6 pts) Trace the following code and write the outputs.

```
main ()
{ int v[] = {9,8,7,6,5,4,3,2,1,0};
  for (int i=0; i<10; i++)
     cout << " " << v[i] << endl;
  bool changed;
  do{changed = false;
     for (int i=0; i<10; i++)
        {if ( v[i] > v[i+1] )
           {  double temp = v[i];
              v[i] = v[i+1];
              v[i+1] = temp;
              changed = true;}}
     }while (changed);
  for (int i=0; i<10; i++) cout << " " << v[i]<< endl;
}
```

9 8 7 6 5 4 3 2 1 0
0 1 2 3 4 5 6 7 8 9

Question 6. (10 PTS) Given the partial class definition

```
class MyClass
{    private:
          float a[50];
     public:
          MyClass(void);
          ??? Fun(??? j);  // Single input argument};
```

Implement the member function `Fun` which will allow the user to insert floating numbers to the desired location in the floating array `a`. Array indices greater than 49 will be inserted in the last location and negative indices will be inserted in the 1st location in the array. Any index between 0-49 will be inserted in the corresponding location in the array. This member function will take only a SINGLE input argument. Comment on each line of your code.

SOLUTION:

In this question your function should be able to put a floating number `x` in the array location `y`

1st method: Use references

```
float& MyClass::Fun(int j)
{
     int index;

     index = j;
     if (j < 0)
          index = 0;
     if(j > 49)
          index = 49;

     return a[index];
}
```

An example use of this function in a main program is

```
MyClass A;
int x = 10;
float y = 1.234

A.Fun(x) = y;
```

2nd method: Use another class definition

```
class NewClass
{
     private:
          float number;
          int index;
```

```
     public:
          NewClass(float a, int b); // initializes number          with
                                    a, index with b
          float GetNumber(void); // returns number
          int GetIndex(void); // returns index
}

void MyClass::Fun(NewClass j)
{
     int index;
     int temp;

     index = temp = j.GetIndex();
     if (temp < 0)
          index = 0;
     if(temp > 49)
          index = 49;

     a[index] = j.GetNumber();
}
```

Then an example usage in main is

```
MyClass A;
int x = 10;
float y = 1.234;
NewClass B(y,x);

A.Fun(B);
```

Question 7. Given the `main` program and the function `Fun`

```
void Fun(int *a, int *b)
{    a = a + 5;
     b = a - 2;};

main()
{    int a[50], b[50];
     int *p, *q;
     p = a + 3;
     q = b + 5;
     Fun(p,q);  // Line (X)}
```

a) **(5 PTS)** What are the contents of `p` and `q` after the execution of line (X)?

SOLUTION:

a) Since call by value is performed the contents of `p` and `q` are unchanged.

b) **(5 PTS)** Propose a way how `Fun` should be modified in order to make `p` point to the first element in `a` and `q` point to the first element in `b`. Also indicate how `Fun` should be called in `line(X)`.

SOLUTION:

b)

1st method:

```
void Fun(int &*a, int &*b)
{
     a = a - 3;
     b = b - 5;
}
```

In line (X):

```
Fun(p,q);  // Line (X)
```

2nd method:

```
void Fun(int **a, int **b)
{
     *a = *a - 3;
     *b = *b - 5;
}
```

In line (X):

```
Fun(&p,&q);  // Line (X)
```

Question 8. (20 pts) Complete the output of the code given below.

```
01  #include  <iostream.h>
02  #include  <stdlib.h>
03
04  int func1(int q[],  int n=4, int m=0)
05  {
06  int  j;
07  for(j=0; j<n; j++) cout<<q[j]+m<<",";
08  cout<<endl;
09  return *(q+m+n+1);
10  }
11
12  int* func2(int *q[], int n=4, int m=0)
13  {
14  int  j;
15  for(j=0; j<n; j++) cout<<*(q[j]+m)<<",";
16  cout<<endl;
17  return *(q+m+n+1);
18  }
19
20
21  int** func3(int **q[],int n=4, int m=0)
22  {
23  int  j;
24  for(j=0; j<n; j++) cout<<**(q[j]+m)<<",";
25  cout<<endl;
26  return *(q+m+n+1);
27  }
28
29  void main()
30  {
31  int i, a, A[100];
32  int *p, *P[100], **r, **R[100];
33  for(i=0; i<100; i++)
34  { A[i]=i+i;
35    P[i]=&A[i]+i+1;
36    R[i]=&P[i]+i+1;
37  }
38
39  cout<<endl<<"line 40:"<< endl;
40  a=func1(A,5,3); cout<<"out:"<<a<<endl;
41  cout<<endl<<"line 42:"<< endl;
42  p=func2(P,4,2); cout<<"out:"<<*p<<endl;
43  cout<<endl<<"line 44:"<< endl;
44  r=func3(R,4,2); cout<<"out:"<<**r<<endl;
45  cout<<endl<<"line 46:"<< endl;
46  a=func1 (A+*P[2]);
47  cout<<endl<<"line 48:"<< endl;
48  func1 (A+*(*R[1]+3));
49  cout<<endl<<"line 50:"<< endl;
50  func1 (func2(R[3]));
51  cout<<endl<<"line 52:"<< endl;
52  func1 (func2(func3(R+1)));
53  }
```

```
OUTPUT:

line 40:
3,5,7,9,11,
out: 18


line 42:
6,10,14,18,
out: 30


line 44:
14,22,30,38,
out: 62


line 46:
20,22,24,26,


line 48:
40,42,44,46,


line 50:
30,34,38,42,
50,52,54,56,



line 52:
14,22,30,38,
54,58,62,66,
74,76,78,80,
```

STACKS QUEUES

Question 1) (25 pts)

You are given the Stack and Queue classes covered in lectures:

```
const int MaxStackSize=50;
template <class T>
class Stack
{
private:
T stacklist[MaxStackSize];
int top;
public:
Stack(void); // constructor
to initialize top
void Push(const T& item);
T Pop(void);
void ClearStack(void);
int StackEmpty(void) const;
int StackFull(void) const;
};
template <class T>
class Queue

{
private:
int front, rear, count;
T qlist[ MaxQSize] ;
public:
Queue(void);
void Qinsert(const T& item);
T QDelete(void);
int QLength(void) const;
int QEmpty(void) const;
};
```

Part a) 12 points
Write a function

```
void findinQ (int key, Queue <int>& MyQ)
```

which searches a given key in a Queue and deletes it if it exists. If the key does not exist the function ends after the search without performing an operation on the queue. You can only use stacks and at most one temporary variable in this function.

```
SOLUTION:
void findinQ (int key, Queue <int>& MyQ)
{
Stack<int> S1;
Stack<int> S2;
int temp;
while(!MyQ.QEmpty())
{
temp=MyQ.QDelete();
if(temp!=key)
S1.Push(temp);
}
while(!S1.StackEmpty())
S2.Push(S1.Pop());

while(!S2.StackEmpty())
MyQ.QInsert(S2.Pop());

}
```

Part b) 13 points You are given the following piece of code. Write OUTPUT1 and OUTPUT2 in the respective boxes.

```
#include "stack.h"
#include "queue.h"
void main ()
{
int i;
Queue<int> Q1,Q2;
Stack<int> S1,S2;
int Arr[6];
for(i=0;i<5;i++)
{Q1.QInsert(i);
Q2.QInsert(i*2);}
i=0;
while(!Q1.QEmpty())
{
Arr[i]=Q1.QDelete();
if(Arr[i]<3)
Arr[i]=Arr[i]*2;

S1.Push(Arr[i]);
S2.Push(Q2.QDelete());

if(Arr[i]>3)
Arr[i]=S1.Pop()+S2.Pop();
i++;
}
while(!S1.StackEmpty())
S2.Push(S1.Pop());

//OUTPUT1
cout<<"stack:";
while(!S2.StackEmpty())
cout<<S2.Pop()<<" ";
cout<<"\n";
//OUTPUT2
cout<<"array:";
for(i=0;i<5;i++)
cout<<Arr[i]<<" ";
}
```

OUTPUT1:

stack:0 2 3 6 2 0

OUTPUT2:

array:0 2 8 3 12

Question 2 (25 pts.) You are given the following `Stack` class definition, implemented as discussed in class:

```
MaxSize = 50; //constant capacity
template <class T>
class Stack
{private:
     T slist [MaxSize];
     int top;
 public:
     Stack(void);
     void Push (const T &item);
     T Pop (void);
     int Stack_Empty (void) const;
     int Stack_Full (void) const;
}
```

Now, you are required to implement a `Queue` class using ONLY stacks (i.e., objects that belong to the class given above,) to store the data items. That is, you can NOT use an array to store the items in the queue, you have to use a stack for that purpose.

(a) (5 pts.) Complete the private part of the following template-based `Queue` class definition:

```
template <class T>
class Queue
{private:

     ..............................
 public:
     Queue (void);
     void QInsert (const T &item);
     T QRemove (void);
     int QEmpty (void);
     int QFull (void);
}
```

(b) (15 pts.) Give the implementation of the `QInsert` and `QRemove` member functions of the `Queue` class, to insert an item at the rear of a queue and to remove the item at the front of the queue, respectively.

(c) (5 pts.) Give the implementations of the `QEmpty` and `QFull` member functions of the `Queue` class, to test whether the queue is empty or full, respectively.

SOLUTION

(a)

```
template <class T>
class Queue
{private:
     Stack<T> SLIFO, SFIFO;
 public:
     Queue (void);
     void QInsert (const T &item);
     T QRemove (void);
     int QEmpty (void);
     int QFull (void);
}
```

(b) Solution idea: Insert items into the LIFO stack; whenever a removal is required (i.e. QRemove called) simply reverse the LIFO into another stack called FIFO, remove from the top, and then restore the LIFO stack.)

```
     void QInsert (const T &item)
     {if SLIFO.Stack_Full()
          {cerr<<"Queue overflow"<<endl; exit(1);}
      Push.SLIFO(item); //insert directly to SLIFO
     }

     T QRemove (void)
     {T temp;
      if SLIFO.Stack_Empty()
          {cerr<<"Queue empty"<<endl; exit(1);}
      while (!SLIFO.Stack_Empty( )) //reverse order
          SFIFO.Push(SLIFO.Pop());
      temp = SFIFO.Pop( ); //remove the first in item
      while (!SFIFO.Stack_Empty())
          SLIFO.Push(SFIFO.Pop()); //restore order
      return temp;
     }
```

(c)

```
     int QEmpty (void)
     return SLIFO.Stack_Empty();

     int QFull (void)
     return SLIFO.Stack_Full();
```

Question 3. (25 pts.)

Write a C++ function, `unbalanced_string`, which checks for balancing of, parenthesis (‘(’,‘)’), brackets (‘[’, ‘]’) and braces (‘{’, ‘}’) in a given string of arbirary characters. For a string to be balanced, every right brace, bracket and parenthesis must correspond to its left counterpart. For example, the sequence [()] is balanced but [(]) is not. The function

```
int unbalanced_string(char *s)
```

will return the result “-1” if the string is balanced or the “index” of the first unmatched left character ‘(’, ‘[’ or ‘{’ in the string if it is unbalanced. In case of a missing left character, the index of the first unmatched right character will be returned. You are allowed to use only the stack data structure developed in the class for this purpose. Two examples of balanced and unbalanced input strings are {x(y[wz])} and {x(y[wz]}, for which, `unbalanced_string` returns -1 and 3, respectively.

SOLUTION:

```
int unbalanced_string(char *s)
{ Stack<char> Sc;
  Stack<int> S1, S2, S3;
  char c; int i=0; int x,y,z;
  while (*s != NULL)
  { if (*s = '(') {Sc.Push(*s); S1.Push(i)}
    else if (*s = '[') {Sc.Push(*s); S2.Push(i)}
    else is (*s = '{') {Sc.Push(*s); S3.Push(i)}
    /* else statements in the following three blocks check
       the invalidity case such as "[(])" */
    else if (*s = ')') {
             if S1.Empty() return i
             else {c=Sc.Pop();
                   x=S1.Pop();
                    if (c='(')
                    else return x}
         }
    else if (*s = ']') {
             if S2.Empty() return i
             else {c=Sc.Pop();
                   x=S2.Pop();
                    if (c= '[')
                    else return x}
         }
    else if (*s = '}') {
             if S3.Empty() return i
             else {c=Sc.Pop();
                   x=S3.Pop();
                    if (c= '{')
                    else return x}
         }
    }
    s++; i++;
  }
  /* now check unbalanced case. If so, return the index of the
     first unmatched character */
  if (S1.Empty() && S2.Empty() && S3.Empty()) return -1
  else {
    x = y = z = Infinity;   /* a very large number */
    while !S1.Empty() x = S1.Pop();
    while !S2.Empty() y = S2.Pop();
    while !S3.Empty() z = S3.Pop();
    /* find the minimum index */
if (x<=y && x<=z) return x
    else if (y<=x && y<=z) return y
    else if (z<=x && z<=y) return z
    else cout << "error"
  }
}
```

Question 5. (25 pts) The following Stack and Queue classes are given:

STACK CLASS	QUEUE CLASS

const int MaxStackSize=50;
template <class T>
Class stack
{private:
int top; T stacklist [MaxStackSize];
public:
Stack(void);//constructor
//stack modification operations
void Push(const T& item); //insert at top
T Pop(void); // delete from top
void SClear(void);
//stack access
T Peek(void) const; // return top item
// stack test methods
int SEmpty(void) const;
int SFull(void) const;
};
	const int MaxQSize=50;
	template <class T>
	class Queue
	{private:
	int front, rear, count; T Qlist [MaxQSize];
	public:
	Queue(void); //constructor
	//queue modification operations
	void QInsert(const T& item); // insert at rear
	T QDelete(void); // delete from front
	void QClear (void);
	// queue access
	T QFront(void) const; // return front item
	// queue test methods
	int QLength(void) const;
	int QEmpty(void) const;
	int QFull(void) const;
	};

a) (14 pts) Complete the implementations of the following two functions which are not members of the given classes (i.e. these are NOT member functions but global functions):

```
template <class T>
T RetrieveElement (Stack<T> &S, int i)
/* returns the i'th element counted from the top of Stack S,
   return value is undefined if i is out of bounds of stack */
```

```
{Stack<T> temp; T tempel;// temporary
     storage
 for (int j=0; !S.SEmpty()&& j<>i; j++)
     temp.Push(S.Pop());
 if j == i tempel = temp.Peek( );
 while (!temp.SEmpty( ))
     S.Push(temp.Pop());
 return tempel;
}
```

Part (a) continued on reverse →

➔ **Question 5. part (a) continued:**

```
template <class T>
void StoreElement(Stack<T>& S, int i, const T& item)
/*stores given item i at the i'th location from stack top,
  undefined items of type T pushed into stack at previously
unused locations
*/
{T UNDEF;//contents undefined
 Stack<T> temp; // temporary storage
 for (int j=0; !S.SEmpty() && j<i; j++)
     temp.Push(S.Pop());
 while (j<i) {temp.Push(UNDEF);j++};
 //at this point j=i
 S.Push(item);
 while (!temp.SEmpty())
     S.Push(temp.Pop());
 return;
}
```

b) (8 pts) Complete the implementation of the following function to insert an element, not at the rear, but at the front of a given queue, using only the public methods of the given queue class, and an extra queue for temporary storage:

```
template <class T>
void QInsertFront (Queue<T>& Q, const T& item)
{Queue<T> temp; //temporary storage
 while (!Q.QEmpty())
         temp.QInsert(Q.QDelete());
  Q.QInsert(item); //store item at final place
  while (!temp.QEmpty()) //restore original
         Q.QInsert(temp.QDelete());
}
```

c) (3 pts) Write the O(.) complexity of the operations in part(a) and in part(b), if the original stack and queue contains n items:

(a) RetrieveElement:	StoreElement:	(b)Q InsertFront:
O(n)	O(n)	O(n)

Question 6.

(4 pts) You are given the following piece of code. Write OUTPUT1 and OUTPUT2 in the respective boxes.

```
#include "stack.h"
#include "queue.h"
void main ()
{
int i;
Queue<int> Q1,Q2;
Stack<int> S1,S2;
int Arr[5];
for(i=0;i<5;i++)
{Q1.QInsert(i);
Q2.QInsert(i*2);}
i=0;
while(!Q1.QEmpty())
{
Arr[i]=Q1.QDelete();
if(Arr[i]<3)
Arr[i]=Arr[i]*2;

S1.Push(Arr[i]);
S2.Push(Q2.QDelete());

if(Arr[i]>3)
Arr[i]=S1.Pop()+S2.Pop();
i++;
}
while(!S1.StackEmpty())
S2.Push(S1.Pop());

//OUTPUT1
cout<<"stack:";
while(!S2.StackEmpty())
cout<<S2.Pop()<<" ";
cout<<"\n";
//OUTPUT2
cout<<"array:";
for(i=0;i<5;i++)
cout<<Arr[i]<<" ";
}
```

OUTPUT1:

stack:0 2 3 6 2 0

OUTPUT2:

array:0 2 8 3 12

Question 7

Top

S:

1	2	3	4		5	6	7		8	9		0	6

Q:

+	-	%	*	/	x	y	z

Front Rear

a) Write the output generated when the code given below is run assuming that the initial contents of S and Q is as given above.

```
Stack<char> S1;
char c;
while (!S.Empty())
{while (!S.StackEmpty())&&(!S.Peek()==””)
    {S1.Push(S.Pop())}
 if (!S.StackEmpty())
    {c=S.Pop()}
 if (!Q.QueueEmpty())&&((!S.StackEmpty())
     {S1.Push(Q.Delete()}
}
while (!S1.StackEmpty())
{cout<< S1.Pop()}
```

Output:

b) Repeat (a) for the code given below assuming that the initial contents of S and Q is as given above.

```
Stack<char> S1, S2;
Queue<char> Q1;
while (!S.StackEmpty()
    {S1.Push(S.Pop())}
while (!S1.StackEmpty())
{ while (!S1.StackEmpty())&&(!S1.Peek()==””)
    {S2.Push(S1.Pop())}
  if (!S1.StackEmpty())
    {c=S1.Pop())
  if (!Q.QueueEmpty())
    {S2.Push(Q1.Delete())}
  while (!S2.StackEmpty())&&(!S2.Peek()==””)
    {Q1.Insert(S2.Pop())}
}
while (!Q1.QueueEmpty())
    {cout<< Q1.Delete()}
```

Output:

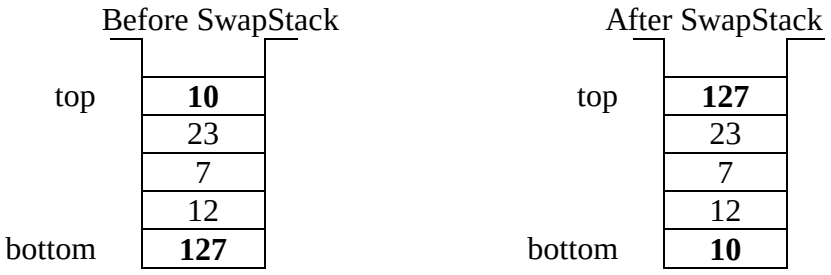
Question 8

You are given the Stack and Queue Class Declarations:

```
template <class T>
class Stack {
private:
T stacklist[MaxStackSize],
int top;
public:
Stack(void);
void Push(const T& item);
T Pop(void);
void Clearstack(void);
T Peek(void) const;
int StackEmpty(void) const;
int StackFull(void) const;
}
```

```
template <class DataType>
class Queue {
private:
int front, rear, count;
DataType qlist[MaxQSize];
public:
Queue(void);
Qinsert(const Datatype item);
DataType Qdelete(void);
void ClearQueue(void);
DataType QFront(void) const;
int QLength(void) const;
int QEmpty(void) const;
int Qfull(void) const;
}
```

Notice that all the data members in these classes private, that is they can not be accessed outside the classes. You are required to write the function SwapStack which will swap the content of the top and bottom elements of a stack (by following steps a and b below). An example case before and after this fuction is called is given below.



a) Complete the following code so that the SwapStack function will operate properly. You are not allowed to declare any other data structures but only those given in the fuction below.

```
Template <Class T>
void SwapStack(Stack<T> myS)
{ T topItem, bottomitem;
   Queue<T> myQ;
   // get topItem from the Stack
   topItem=myS.Pop();
   // move the rest of the myS into myQ
   while ...
   { ...
     };
   // load the bottomItem appropriately
   bottomItem=...
   // other operation(s)between myS and myQ if any necessary



   ...



   // push  either topItem or bottomItem into myS
   // whichever appropriate
   ...
   // push other items into myS by taking from myQ
   while ...
   { ...
     };
   // push  either topItem or bottomItem into myS
   // whichever appropriate
   ...
}
```

b) Now assume that the data member top and stacklist are also public and repeat (a)

```
Template <Class T>
void SwapStack(Stack<T> myS)
{ T ...
   topitem= ...

   ...

   ...                  =topitem;
}
```

SOLUTION: You are given the Stack and Queue Class Declarations:

```
template <class T>
class Stack {
private:
T stacklist[MaxStackSize],
int top;
public:
Stack(void);
void Push(const T& item);
T Pop(void);
void Clearstack(void);
T Peek(void) const;
int StackEmpty(void) const;
int StackFull(void) const;
}
```

```
template <class DataType>
class Queue {
private:
int front, rear, count
DataType qlist[MaxQSize];
public:
Queue(void);
Qinsert(const Datatype item);
DataType Qdelete(void);
void ClearQueue(void);
DataType QFront(void) const;
int QLength(void) const;
int QEmpty(void) const;
int Qfull(void) const;
}
```

Notice that all the data members in these classes private, that is they can not be accessed outside the classes. You are required to write the function SwapStack which will swap the content of the top and bottom elements of a stack.

a) Complete the following code so that the SwapStack function will operate properly. You are not allowed to declare any other data structures but only those given in the fuction below.

```
Template <Class T>
void SwapStack(Stack<T> myS)
{ T topItem, bottomitem;
   Queue<T> myQ;
   // get topItem from the Stack
   topItem=myS.Pop();
   // move the rest of the myS into myQ
   while !(myS.StackEmpty())
   { myQ.Qinsert(myS.Pop());}
   // load the bottomitem appropriately
   bottomItem=myQ.Qdelete();
   // solutions setting Bottom item previously or later
   // also accepted
   //
   // write here other operation(s)between myS and myQ
   // if there is any that you find necessary
   while !(myQ.QEmpty())
   { myS.push(bottomItem);bottomItem=myQ.Qdelete()};
   while !(myS.StackEmpty())
   { myQ.Qinsert(myS.Pop())};
   // push topItem into myS
   myS.Push(topItem);
   // push other items into myS by taking from myQ
   while !(myQ.QEmpty)
   myS.Push(myQ.Qdelete());
   // push bottomItem into myS
   myS.Push(bottomItem);
}
```

b) Now assume that the data member top and stacklist are also public and repeat (a)

```
Template <Class T>
void SwapStack(Stack<T> myS)
{ T topitem;
   topitem=myS.Peek();
   myS.stacklist[top]= myS.stacklist[0];
   myS.stacklist[0]= topitem;
}
```

COMPLEXITY

Question 1 (25 pts)

a)Given the following functions covered in class:

```
template <class T>
int SequentialSearch (T list[ ], T key, int N);
template <class T>
int BinarySearch (T list[ ], int low, int high, T key);
```

and the following function which returns a random integer with O(1) complexity.

```
int RandInt ();
```

Function `RandomSearch` performs a search to find a given key in a sorted array of N items. `RandomSearch` is defined as follows:

```
template <class T>
int RandomSearch (T list[ ], int low, int high, T key, int N)
{
int index;
int x=RandInt();
if(x%2==0)//x mod 2 operation
index= SequentialSearch (list[ ], key, N);
else
index=BinarySearch (list[ ], 0, N-1, key);
return index;
}
```

Find the complexity of `RandomSearch` in O(.), Ω(.) and Θ(.). Justify your results. If you cannot express the complexity in any of these forms, state your reasons.

Sequential Search : O(N), Ω(1)
Binary Search : O(logN), Ω(1)
Random Search: O(N), Ω(1), No Θ(.).

b) Consider a two dimensional array that holds *N* integers. The array has a fixed column size of *K*. For each row *i*; *K* random integers are picked between min_i and max_i such that $min_i > max_{(i-1)}$ and $max_i < min_{(i+1)}$ and stored in the array.

You are given the following algorithm to search for a given integer key for such a two dimensional array:

For each row *i*, the algorithm checks if the key is between min_i and max_i. If that is the case, then it compares the key with all items in row *i* until it finds a match. If there is no match it exits without a match.

Find the complexity of this search algorithm in O(.), Ω(.) and Θ(.). Justify your results. If you cannot express the complexity in any of these forms, state your reasons.

Number of rows in the array=N/K
N/K comparisons to find the row and then K comparisons to find the item in the correct row.
K is a constant.
Complexity=O(N/K+K)
Complexity=O(N)
Ω(1), No Θ(.).

Question 2
Determine, showing your arguments briefly, the $O(\ )$, $\Omega(\ )$ and $\Theta(\ )$ complexities of:

(a) The function $(n^3 + 7) / (n + 1)$

(b) The running time of the following *C++* code, as a function of n :

```
for (i = 1; i < n; i ++)
        for j = 1; j < i * i; j ++))
                count ++;
```

(Note that $1^2 + 2^2 + 3^2 + ... + n^2 = n (n + 1) (2n + 1) / 6$)

(c) The worst case running time of the following algorithm, described informally, as a function of n :

Search a sorted array of n elements for a given *key*:

1. Start with size ← full table size.
2. Partition the table to be searched into three.
   If the table is too small for this (i.e., if size < 3) simply check its contents and exit either with success or with failure.
3. If *key* < the last element of the lowest partition, from now on, consider only this lowest partition and go back to step 2. (In steps 3 and 4, if the key is found at the comparison operation, exit the algorithm with success.)
4. If *key* < the last element of the middle partition, from now on, consider only this middle partition and go back to step 2.
5. Otherwise from now on, consider only the top partition and go back to step 2.

Solution:

(a) $f(n) = n^2 - 1 + (8 / (n + 1))$ so $f(n) = O(n^2), \Omega(n^2)$ and $\Theta(n^2)$

(b) The inner loop is executed

$1^2 + 2^2 + 3^2 + ... + n^2$ times, so

$f(n) = n (n + 1) (2n + 1) / 6 = O(n^3), \Omega(n^3)$ and $\Theta(n^3)$

(c) In each step, the table of n items is partitioned into 3
This can be done at most $\log_3 n$ times.
The exit operation is of constant complexity
(at most three comparisons, regardless of total table size,)
So, total worst case complexity is $O(\log n)$.

(Note: "best" case is of course $\Omega(1)$ corresponding to success at the first comparison, hence $\Theta(\ )$ is not defined.)

Question 3. (30 pts) Show all your reasoning and work.

a) (6 pts) Let $f(n) = n$, $g(n) = n^{(1+\sin(n))}$ and $h(n) = n^2$.

Is $f(n) = O(g(n))$ correct?

Is $f(n) = \Omega(g(n))$ correct?

Is $h(n) = \Omega(g(n))$ correct?

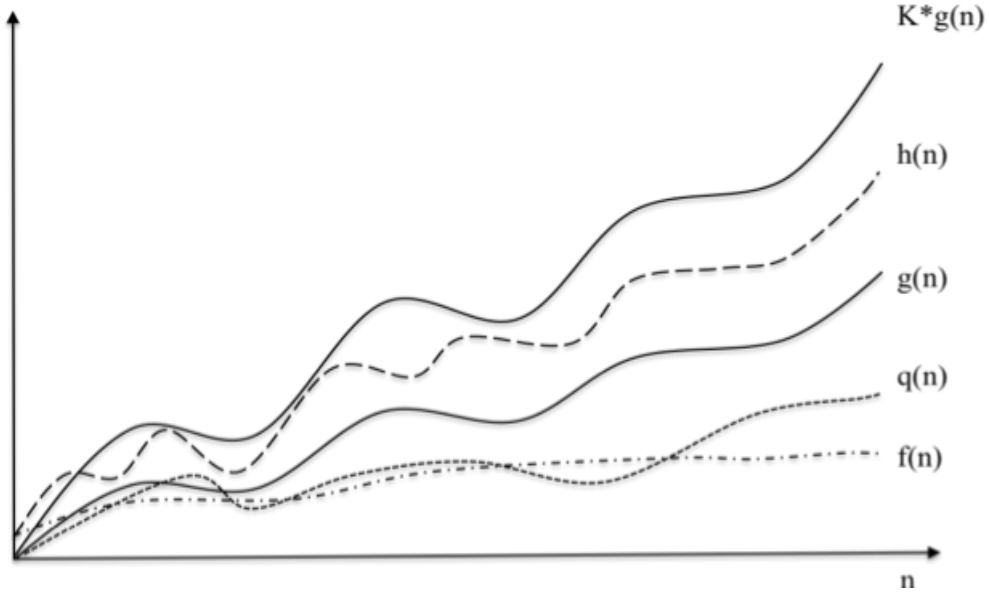
Explain.

Solution:

First two statements are incorrect since $g(n) = n^{(1+\sin(n))}$ changes between 1 and n^2.

The third statement is correct.

b) (6 pts) Consider the given figure. Let $p(n) = h(n) + f(n)$



Find the TIGHTEST O(.) ,Ω(.) and Θ(.) complexities of $h(n)$, $q(n)$ and $p(n)$ expressed in terms of $f(n)$ and $g(n)$.

Solution:

$h(n)$: $\Theta(g(n))$

$q(n)$: $O(g(n))$, $\Omega(f(n))$

$p(n)$: $\Theta(g(n))$

c) (18 pts) Consider two arrays of integers A1 and A2. Both A1 and A2 store N distinct integers where $N=2^k-1$, k is an integer. The contents of A1 and A2 are independent of each other. A2 is sorted from smallest to largest. You are required to find how many integers in A1 also exist in A2 using the binary search algorithm covered in the lectures.
What is the complexity of this task in terms of the TIGHTEST O(.) ,Ω(.) and Θ(.)?
Hint1: Consider the best and worst cases. *Construct an example for the best case.*

Hint2: $\sum_{i=1}^{m} i2^i = 2(1+(m-1)2^m)$

Answer:
Complexity in O(.) : $N\log N$
Complexity in Ω(.): $N\log N$
Complexity in Θ(.): $N\log N$

Solution:
Worst case: N unsuccessful searches, each terminates in $\log N$. Complexity in O(.) :$N\log N$
Best case: All N items are found with the shortest number of trials. Consider the following example:
A1=[4,2,6,1,3,5,7], A2=[1,2,3,4,5,6,7]
Search for 4:1 cut in the middle, found.
Search for 2 and 6:1 cut in the middle, not found, 1 more cut, found.
Search for 1,3,5,7: found after 3 cuts.
You find 1 element with 1 cut, 2 elements with 2 cuts, 4 elements by 3 cuts. Genaralize: You find i elements with 2^{i-1} cuts.
You need a total of $\sum_{i=1}^{k} i2^{i-1} = (1+(k-1)2^k)$ operations for $N=2^k-1$.

Complexity in Ω (.) :$N\log N$
Complexity in Θ(.): $N\log N$

Question 4. (25 pts) Solutions that only consist of expressions O(.) ,Ω(.) and Θ(.) will receive no credit. Show all your reasoning and work.

a) Derive the complexity of the following code segment in O(.) ,Ω(.) and Θ(.):

```
int A[N];
int B[N];
bool found=false;
int i,j;
int key;
cin>>key;//get the key from the user
for(i=0;i<N;i++)
for(j=0;j<N;j++)
if(A[i]==B[j])
{
found=true;
break;//breaks both for loops
}

if(found==true)
BinarySearch (A, 0, N-1, key);
```

Recall: `BinarySearch (int list[ ], int low, int high, int key)`

Worst case: Match is found for A[N-1]==B[N-1]: $O(N^2)$
Then conduct binary searchO(logN).
Total: $O(N^2)+ O(logN)= O(N^2)$

Best case Match is found for A[0]==B[0] then conduct Binary search: Ω(1) if key found at the first checked location.

Θ(.) does not exist.

Note if no match is found, $O(N^2)$ but this is not the complexity

b) Consider a stack with N items.
What is the O(.) complexity of accessing the most recently inserted item in the stack.
What is the O(.) complexity of accessing oldest item in the stack.

O(1): it is the top item
O(N): it is the bottom item, all other items have to be taken out first

c) What is the O(.) complexity of searching for a given key in a stack or a queue ?

O(N): Both of these data structures provide sequential access.

Question 5 (25 pts)
a) Given the following function:

```
void DoSomething (int N);
{
int i=0;
int j=0;
while(i<N)
{
statement x;
      while(i<N)
      {i++;
      statement y;}
      j++;
      statement z;
}

}
```

The complexity of statements are given as:
`statement x:` O(log*N*)
`statement y:` O(*N*)
`statement z:` O(1)

What is the complexity of `DoSomething` in O(.)?

b)You are given the function
`int Diff(int i,int j)` which returns the absolute value of i-j.

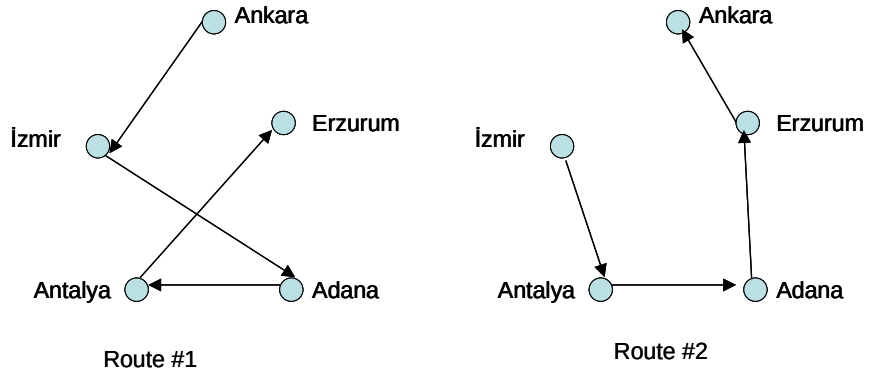
Consider the function `FindMinDiff` below which takes an array `A` of size `N` and returns the minimum absolute difference between any two integers in `A`.

```
int FindMinDiff ( int A[], int N)
{
int i,j,diff;
int MinDiff=Diff(A[0],A[1]);
for(i=0;i<N;i++)
for(j=i+1;j<N;j++)
{
diff=Diff(A[i],A[j]);
if(diff<MinDiff)
MinDiff=diff;
}
return MinDiff;

}
```

What is the complexity of `FindMinDiff` in O(.),Ω(.) and Θ(.)?

c)Traveling Salesman Problem:
A salesman wants to visit N towns to sell books. He visits every town exactly once. He can start from any town.
Example:
For 5 towns, two possible routes are given below.



Let F(N) be the function that computes the route with the shortest possible route for the traveling salesman. What is the complexity of F(N) in O(.) notation.

Solution:

a)

```
void DoSomething (int N);
{
int i=0; ➔ ti
int j=0;
while(i<N)➔ ta
{
statement x; ➔tx
      while(i<N)➔ ta
      {i++; ➔tb
      statement y;}➔ ty
      j++; ➔ tc
      statement z; ➔tz
}

}
```

Complexity: ti+ta+tx+N(ta+tb+ty)+tc+tz=tA+tx+N(tB+ty)+tz➔
tA+O(logN)+N(tB+O(N))+O(1)=tC+O(logN)+NtB+NO(N)=O(N^2)

b)

```
int FindMinDiff ( int A[], int N)
{
int i,j,diff;➔ti
int MinDiff=Diff(A[0],A[1]); ➔tD
for(i=0;i<N;i++)➔ta1,ta2,ta3
for(j=i+1;j<N;j++)➔tb1,tb2,tb3
```

```
{
diff=Diff(A[i],A[j]); ➔tD
if(diff<MinDiff) ➔tc
MinDiff=diff; ➔te
}
return MinDiff; ➔tr

}


for(j=i+1;j<N;j++)➔tb1,tb2,tb3
==
for(j=0;j<N-i-1;j++)➔tb1,tb2,tb3
```

ti+tD+ta1+(N+1)ta2+Nta3+sum(i=1 to N)(tb1+(N-i)tb2+(N-i-1)ta3)
$O(N^2)$ Omega(N^2), Theta(N^2)

c)
Brute force search: Check all combinations of cities: O(N!)

Question 6) (25 pts)
a) Given the following function:

```
void JustForComplexity (int N, int k)
{
T1(N);
for (int i=0;i<N;i++)
if(k<N)
      T2(N);
else
      {
      T3(N);
      return;
      }

}
```

The complexity of T1, T2 and T3 are given as:
`T1(N):` $\Theta(N)$
`T2(N):` $O(N\log N)$
`T3(N):` $\Theta(N^2)$

Derive the complexity of `JustForComplexity` in O(.)

b) 4 algorithms A, B, C and D must be executed on a set of N items to produce a certain result R. At time t=0 algorithms A, B and C start to run in parallel and finish at times tA, tB and tC respectively. R is correct only when tA<tB<tC order is satisfied. Algorithm D starts to run when algorithm C is finished.
The known information about the complexities of the algorithms is as follows:
Algorithm A: $\Theta(N)$
Algorithm C: $\Theta(N^3)$
Algorithm D: $O(2^N)$
i)What should be the complexity of Algorithm B in O(.) and ,Ω(.) such that R is correct.
ii)What is the overall complexity of producing the correct result in O(.).

Solutions that only consist of expressions O(.), Ω(.) will receive no credit. Also explain your reasoning in deriving them

Solution:
a) `if(k<N): T1(N) and N.T2(N) will be executed.`
`O(N) + N.O(NlogN)=>` $O(N^2\log N)$ `is the dominant term`
`if(k>=N): T1(N) and T3(N) will be executed.`
`O(N) + O(`N^2`)=>` $O(N^2)$ `is the dominant term`
`Over all: Max(O(`$N^2\log N$`), O(`N^2`)) =` $O(N^2\log N)$

b)
i)Algorithm B must be finished after A. So the shortest execution time of B must be longer than the longest execution time of A. Algorithm B must be Ω(f(N)) such that f(N)>N as N goes to infinity. One possible choice is $\Omega(N^2)$. Algorithm B must be finished before C. So the longest execution time of B must be shorter than C. Algorithm B must be O(g(N)) such that g(N)< N^3 as N goes to infinity. One possible choice is $O(N^2)$.
ii) $O(2^N)$