Ain Shams University
Faculty of Computer and Information Sciences
Computer Science Department
Final 2017-2018 REMAKE Exam
Prof. Zaki Taha
Dr. Sally Saad

First Year – Second Term
Date: May 7th, 2018
Time: 180 minutes
Course: CSW150
Total: 125 points

Fundamental of Structured Programming Using C++

Please attempt ALL questions, **separate them clearly**, and think before you write.

QUESTION (1) **(40 points)**

1.1 Type True/False in front of each statement and CORRECT the false ones. Use the following template table for your answer.

Answer	Correction of False statements
1.1. True/False	…
1.2. True/False	…

(1) The Compiler is responsible to detect syntax errors. (T)

(2) According to Visual Studio compiler, this code is correct: ***int size=5, myArr[size]={0};*** (F)
const ***int size=5, myArr[size]={0};***

(3) Every recursive function requires a recursive case as well as a base case to stop recursion. (T)

(4) The following code segment is correct.

```
struct Date{ int year; int month; int day;};
struct Person{ char name[10]; Date birthDay;};
void main(){ Person Bill; cin>>Bill.name;}
```

(T)

(5) It's a type mismatch to assign an integer variable to an integer pointer variable. (T)

(6) The statement

```
struct my_struct {int num1, num2; char signs[4];double result;};
```

allocates 20 bytes in memory.

ZERO bytes (F)

(7) To use a built-in function defined in an external library you need just to call the function. (F)
and include the library file

(8) When declaring 2D array we should specify at least the first dimension. (F)
second dimension

(9) Every iterative function can be solved by recursion. (F)
the opposite

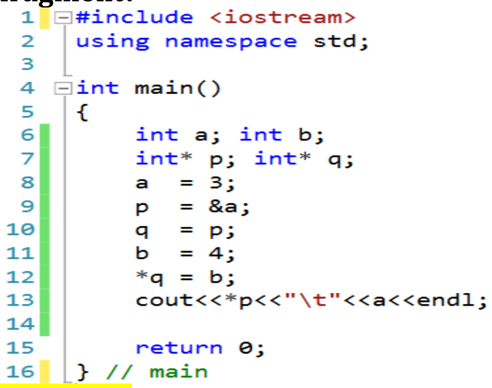
(10) Variables passed in a function call are called formal parameters while the ones declared in function header are called function arguments. (F)
the opposite

QUESTION (2) **(20 points)**

Choose the correct answer (ONLY ONE). (5 points each)
PLEASE FOLLOW THIS ANSWER FORMAT:

Question Number	Answer
3.1	Your choice letter
3.2 …	Your choice letter …

(1) What is the output of the following code fragment?



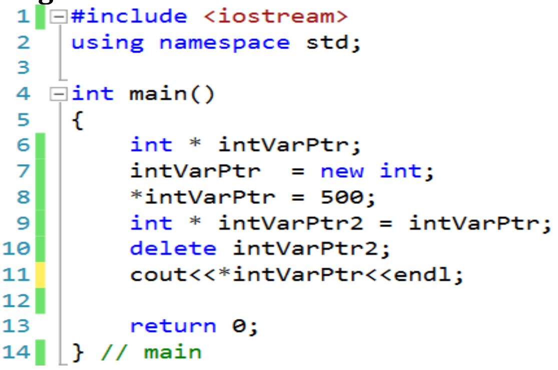
```
#include <iostream>
using namespace std;

int main()
{
	int a; int b;
	int* p; int* q;
	a  = 3;
	p  = &a;
	q  = p;
	b  = 4;
	*q = b;
	cout<<*p<<"\t"<<a<<endl;

	return 0;
} // main
```

(a) 4   4   **(b)** 4   3
(c) 3   3   **(d)** 3   4
(e) *Garbage* 3

(2) What is the output of the following code fragment?



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

int main()
{
	int * intVarPtr;
	intVarPtr   = new int;
	*intVarPtr = 500;
	int * intVarPtr2 = intVarPtr;
	delete intVarPtr2;
	cout<<*intVarPtr<<endl;

	return 0;
} // main
```

(a)500   **(b)** 00000000
(c) Run time Error.   **(d) Garbage.**
(e) Syntax Error.

(3) Where can you NOT declare a variable in a C++ program?

(a) Within the parameter list of a function definition.
(b) Within the block of a function body.
(c) Within the argument list of a function call.
(d) Outside of any function.
(e) None of the above.

(4) Arrays are passed as parameters to functions via

(a) Passing by reference.
(b) Passing by value.
(c) Can use (a) or (b)
(d) We cannot pass arrays as parameters.
(e) None of the above.

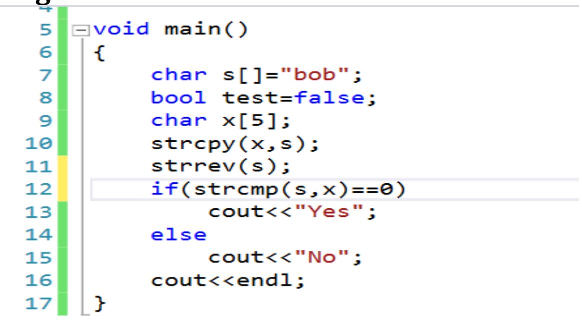
(5) The following statements:
`int a[10] = {1,2}, b[10] = {}; a = b;`

(a) Assigns all values of elements of `a` to elements of `b`.
(b) Assigns non-zero values from elements of `a` to elements of `b`.
(c) Assigns all values of elements of `b` to elements of `a`.
(d) Incorrect expression so no values are assigned.
(e) Makes the two variables point to the same address.

QUESTION (3)   **(25 points)**

Answer the following questions

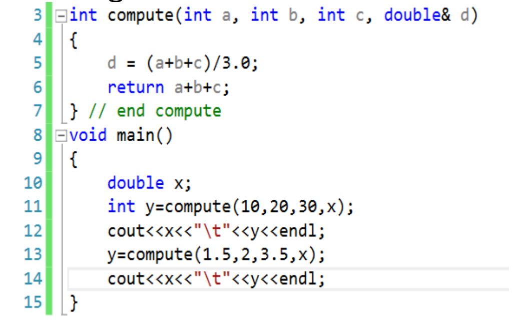
(a) What is the output of the following code fragment?



```
void main()
{
	char s[]="bob";
	bool test=false;
	char x[5];
	strcpy(x,s);
	strrev(s);
	if(strcmp(s,x)==0)
		cout<<"Yes";
	else
		cout<<"No";
	cout<<endl;
}
```

```
Output:YES
```

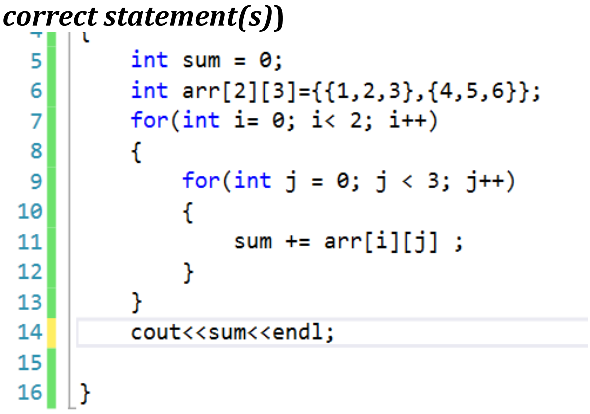
(b) What is the output of the following code fragment?



```
int compute(int a, int b, int c, double& d)
{
	d = (a+b+c)/3.0;
	return a+b+c;
} // end compute
void main()
{
	double x;
	int y=compute(10,20,30,x);
	cout<<x<<"\t"<<y<<endl;
	y=compute(1.5,2,3.5,x);
	cout<<x<<"\t"<<y<<endl;
}
```

```
Output:
20    60
2     6
```

(c) The following code outputs the sum of the 2d array: 21. Modify it to output the sum of row by row: 6 and 15 . *(Write down the line number having the modification and the correct statement(s))*

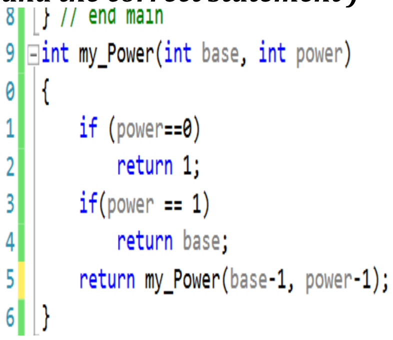


```
5       int sum = 0;
6       int arr[2][3]={{1,2,3},{4,5,6}};
7       for(int i= 0; i< 2; i++)
8       {
9           for(int j = 0; j < 3; j++)
10          {
11              sum += arr[i][j] ;
12          }
13      }
14      cout<<sum<<endl;
15
16  }
```

Modification:

- **line# 8: sum=0;**
- **line# 14 replace with brace in line # 13**

(d) Correct the code error(s) of Power function to be solved recursively. *(Write down the line number having the error(s) and the correct statement)*



```
8  } // end main
9  int my_Power(int base, int power)
0  {
1      if (power==0)
2          return 1;
3      if(power == 1)
4          return base;
5      return my_Power(base-1, power-1);
6  }
```

Modification:

- **line# 5: return base*my_Power(base, power-1);**

(e) Write down the expected output of the following pseudo-code if we input (5, 3,7,2,9)

```
do
 swapped = false
 for i = 0 to indexOfLastUnsortedElement-1
    if leftElement > rightElement
      swap(leftElement, rightElement)
      swapped = true
    ENDIF
 ENDFOR
while swapped
ENDDO
```

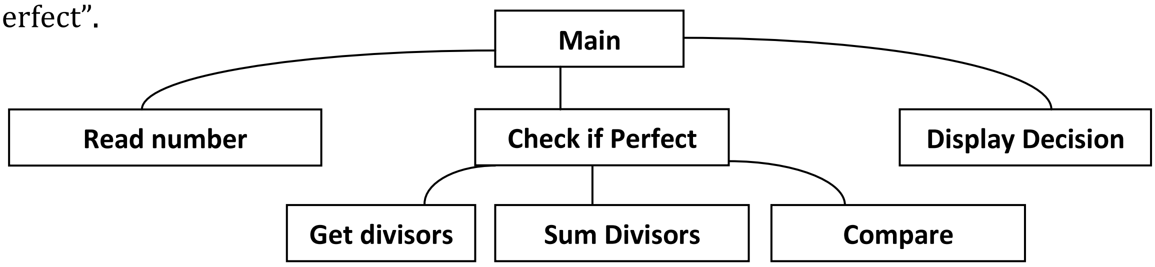
Output:2 3 5 7 9

QUESTION (4) (40 points)

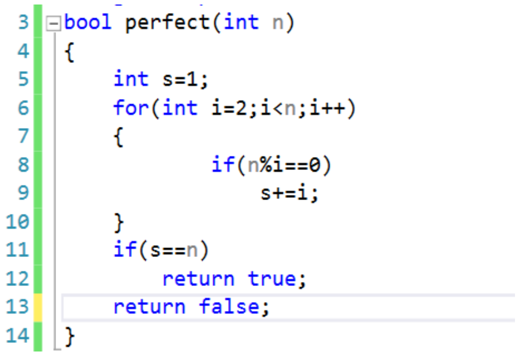
Problem: an integer N is "perfect" if N is equal to the sum of the positive integers K such that K<N and K is a divisor of N.

Example: 6, because 1, 2, and 3 are its proper positive divisors, and 1 + 2 + 3 = 6.

(a) Use **top-down** approach to design a solution to determine if a given integer is "perfect".

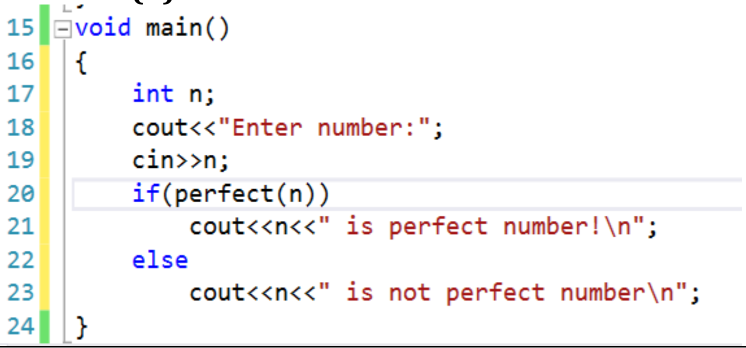


(b) Write the definition of a C++ function `bool perfect(int);` that decides if a given integer is perfect or not.



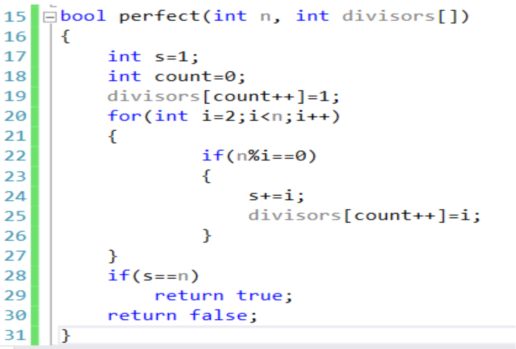
```
bool perfect(int n)
{
    int s=1;
    for(int i=2;i<n;i++)
    {
            if(n%i==0)
                s+=i;
    }
    if(s==n)
        return true;
    return false;
}
```

(c) Write a complete program to solve the Problem using the function ***perfect*** that you just defined in **(b)**.

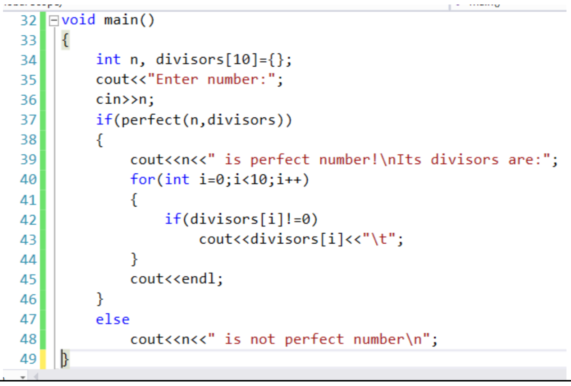


```
void main()
{
    int n;
    cout<<"Enter number:";
    cin>>n;
    if(perfect(n))
        cout<<n<<" is perfect number!\n";
    else
        cout<<n<<" is not perfect number\n";
}
```

(d) Add to your code an overloaded function that returns also the divisors of the given integer.



```
bool perfect(int n, int divisors[])
{
    int s=1;
    int count=0;
    divisors[count++]=1;
    for(int i=2;i<n;i++)
    {
            if(n%i==0)
            {
                s+=i;
                divisors[count++]=i;
            }
    }
    if(s==n)
        return true;
    return false;
}
```



```
void main()
{
    int n, divisors[10]={};
    cout<<"Enter number:";
    cin>>n;
    if(perfect(n,divisors))
    {
        cout<<n<<" is perfect number!\nIts divisors are:";
        for(int i=0;i<10;i++)
        {
            if(divisors[i]!=0)
                cout<<divisors[i]<<"\t";
        }
        cout<<endl;
    }
    else
        cout<<n<<" is not perfect number\n";
}
```

Best Wishes ☺

Prof. Zaki Taha

Dr. Sally Saad