Lab work – functions (passed by reference)

1) Write a program which does the following:

- Ask the user for the radius of a circle.
- Write and call a function to calculate the circumference of a circle given a radius. The prototype (title-line) is as follows:

```
double circumference(double radius);
```

The formula for the circumference of a circle with radius r is $C = 2\pi r$. You can approximate π as 3.14.

Sample I/O #1 (User input is in **bold**)

```
What is the radius of the circle? 5
The circumference is 31.4
```

Sample I/O #2 (User input is in **bold**)

```
What is the radius of the circle? 3.5
The circumference is 21.98
```

```
//Solution 1:
#include <iostream>
using namespace std;

double circumference(double radius) {
  double c = 2 * 3.14 * radius;
  return c;
}

int main() {
  double r;
  cout << "What's the radius? ";
  cin >> r;

  double x;
  //store result in variable x
  x = circumference(r);
  cout << "Circumference: " << x << endl;

  return 0;
}
```

```
//Solution 2:
#include <iostream>
using namespace std;

//prototype, above main
double circumference(double radius);

int main() {
  double r;
  cout << "What's the radius? ";
  cin >> r;

  double x;
  //store result in variable x
  x = circumference(r);
  cout << "Circumference: " << x << endl;

  return 0;
}

//function defined here, below main
double circumference(double radius) {
  double c = 2 * 3.14 * radius;
  return c;
}
```

2) (from p.114 in Schaum's Programming with C++) Write and test the following computeCircle() function that returns the area a and the circumference c of a circle with given radius r:

void computeCircle(double& a, double& c, double r)

```
void computeCircle(double& area, double& circumference, double r) {
   const double PI=3.14;
   area = PI * r * r;
   circumference = 2 * PI * r;
}
```

Use the following to test in main():

```
   double a, c, r;
   cout << "Enter the radius:" << endl;
   cin >> r;
   computeCircle(a, c, r);
   cout << "Area of a circle of radius " << r << " is " << a
           << "\nand its circumference is " << c << endl;
```

3) Write a function called *cube* which takes an integer parameter by **reference**, and cubes the parameter:

```
void cube(int& n) {
   n = n * n * n;
}
```

Use the following to test in main():

```
   int x = 0;
   cout << "Please enter an integer and I will find you its cube: ";
   cin >> x; //If user enters 2 here
   cube(x);
   cout << “The cube is: “ << x << endl; //Program will print 8 on screen
```

Analyze a program for output:

4) (prac2.pdf) Consider the following C++ program.

```
#include <iostream>
using namespace std;
int fun(int &x, int &y) {
      if (y <= 0) return x;
      x = x + 2;
      cout << x << y << endl;
      return x * y;
}
int main() {
      int x = 4, y = 0;
      cout << fun(x, y) << endl;    // line a
      fun(y, x);                    // line b
      fun(x, y);                    // line c
      fun(y, x);                    // line d
      cout << fun(x, y) << endl;    // line e
      return 0;
}
```

What is the output from the program at each of the following lines:

(a) line a: 4
(b) line b: 24
(c) line c: 62
(d) line d: 46
(e) line e:
84
32

5) (prac2.pdf) Consider the following C++ program.

```
#include <iostream>
using namespace std;
int fun(int &x, int y) {
      x = x + 1;
      y = y - 1;
      return y;
}
int main() {
      int x = 2, y = 7, z = 10; string s = "007";
      cout << ((double) y) / x << endl;         // line (a)
      if (!((x > y) && (y > 5))) s = "008";
      cout << s << endl;                        // line (b)
      z %= y; cout << z << endl;                // line (c)
      cout << fun(z, y) << endl;                // line (d)
      fun(x, y); cout << y - x * 2 << endl;     // line (e)
}
```

(a) What is the output at line (a)? 3.5
(b) What is the output at line (b)? 008
(c) What is the output at line (c)? 3
(d) What is the output at line (d)? 6
(e) What is the output at line (e)? 1

Short Blocks of code:

6) (prac2.pdf modification) Write blocks of code to perform the functions used in the following main program. Your blocks must match the given title lines. Each block should be a short function of only a few lines.

```
int main() {
   int a = 3, b = 1, c = 2;
// (a) Swaps values
   swap(b, c);
   swap(b, c);
// (b) Rotate a,b,c so as to print 1,2,3
   rotate(a, b, c);
   cout << a << b << c << endl;
}
```

(a)
```
void swap(int &x, int &y) {
   int temp = x;
   x = y;
   y = temp;
}
```

(b)
```
void rotate(int &x, int &y, int &z) {
   int temp = x;
   x = y;
   y = z;
   z = temp;
}
```

Title Lines:

7) (prac2.pdf & prac3.pdf) Write the best **title lines** for the functions that are called by the following main program. **Do not supply blocks for the functions.**

```
int main() {
      int x = 0, y = 1, z = 2;
      x = sum(z, y);                  // (a) sets x to the sum: 3
      reset(y, z);                    // (b) replaces y by the value of z
      makeNegative(z);                // (c) make z negative
      boost(x, y);                    // (d) increase x by the value of y
      boost(y, mystery(y, z));   // (e) boosts y by a mystery amount
      return 0;
}
```

(a) Title line for sum. `int sum(int z, int y)`

(b) Title line for reset. `void reset(int &x, int y)`

(c) Title line for makeNegative. `void makeNegative(int& x)`

(d) Title line for boost. `void boost(int &a, int b)`

(e) Title line for mystery. `int mystery(int a, int b)`