

ANSWER THE FOLLOWING QUESTIONS.

MAXIMUM DEGREE: 70 POINTS

Q1) Write a program that accepts 100 decimal numbers then compute the average of the entered numbers. Attempt to store the numbers less than the average in another array lessAvg[] and the numbers greater than the average in a third array greaterAvg[]. Finally, display the numbers in both of lessAvg[] and greaterAvg[] arrays and the summation of the numbers in each of them.

(8 points)

Q2.(a) Write a recursive function power(base, exponent) that takes two arguments (a signed double value for the base and a signed integer value for the exponent) and when invoked, returns $\text{base}^{\text{exponent}}$. Attempt to satisfy the following two rules:

(6 points)

- if the exponent ≥ 0 then: $\text{base}^{\text{exponent}} = \text{base} \cdot \text{base}^{\text{exponent}-1}$, and $\text{base}^0 = \text{base}$
- if the exponent < 0 then : $\text{base}^{\text{exponent}} = 1/\text{base} \cdot \text{base}^{\text{exponent}+1}$, and $\text{base}^{-1} = 1/\text{base}$.

(b) Write a C++ program using two nested loops to draw the following shape. All asterisks (*) should be printed by a single statement of the form cout << "*"; inside the loop body.

*

***, ***

*

Q3.(a) Rewrite the following code using switch statement.

(4 points)

```
int m=7, n=4, x= 3;
if(x==1||x==2)
    cout <<m-n;
else if(x==3)
    cout <<m+n;
else
    cout <<m*n;
```

(b) Rewrite the following code using for-loop structure.

(4 points)

10 11 12 13 14

```
int x = 10;
do
{
    do{
        cout << x << " ";
    }while(++x<15);
    cout << endl;
}while(++x <=20);
```

For (x ; x < 20 || x == 20; x++)
{
 cout << x << " ";
}
For (x ; x < 15; ++x)
{
 cout << x << " ";
}

1/3 3

Q4) What is the output of executing the following code?

(16 points)

A	<pre>#include <iostream.h> int main() { int x[4] = { 4,3, 2 }; int* ptr; ptr=x; for(int j=0; j<4; j++) cout << " " <<++*(x+j) * x[j] << endl; return 0; }</pre>	B	<pre>#include <iostream.h> int main() { int x =8; do { x+=2; cout << "\n " <<x; } while(x<20); return 0; }</pre>
C	<pre>#include<iostream.h> void sp(float* J, float* I, int* count) { static int c=0; float n; ~ c++; *count=c; n=j*2; *j=i*2; *i=n; } int main() { float x=5, y=10; char ch; int c; sp(&x, &y, &c); cout << "\n x and y become : "<<x<<" and " <<y; cout << "\n the value of c is: "<<c; sp(&x, &y, &c); cout << "\n x and y become : "<<x<<" and " <<y; cout << "\n the value of c is: "<<c; return 0; }</pre>	D	<pre>#include <iostream.h> int main() { int x=3, y=2, z=1; bool b, c=0; b=x<y==+z>y; cout << "\n " <<b; b=z<-1==x+y>y; cout << "\t " <<b; c= 100*30/2; cout << "\t " <<c; x=++x+x--; cout << "\t " <<x; return 0; }</pre>
E	<pre>#include<iostream.h> int main() { char x='a'; switch (x+2) { case 1+b': cout << "hi"; break; case 'b': cout << "\n the ring"; break; default: cout << "\n halla"; } return 0; }</pre>	F	<pre>#include <iostream.h> #include <cstring> int main() { char str1[] = "Good luck"; int MAX = strlen(str1); char str2[100]; for(int j=1; j<=MAX; j++) str2[MAX-j] = str1[j-1]; str2[MAX]='\0'; cout << str2 << endl; return 0; }</pre>
G	<pre>#include <iostream.h> class emp { private: static int cnt; public: emp() { cnt++; } int getcnt() { return cnt; } }; int emp::cnt = 0; int main() { int x; emp f1, f2; cout << "\n " << f1.getcnt() << "\t" << f2.getcnt(); emp f3, f4; cout << "\n " << f1.getcnt() << "\t" << f4.getcnt(); return 0; }</pre>	H	<pre>#include <iostream.h> void repchar() { for(int j=0; j<4; j++) cout << "*"; cout << endl; } void repchar(char ch) { for(int j=0; j<4; j++) cout << ch; cout << endl; } void repchar(char ch, int n) { for(int j=0; j<n; j++) cout << ch; cout << endl; } int main() { repchar(); repchar('+', 7); repchar('='); return 0; }</pre>

Q5) State which of the following are true and which are false.

(10 points)

- (a) An array can store many different types of values.
- (b) If there are fewer initializers than the number of elements in the array, the remaining elements are automatically initialized to the last value in the list of initializers.
- (c) "\n" Position the screen cursor to the beginning of the current line.
- (d) In a class definition, data or functions designated private are not accessible to member functions of that class.
- (e) Data items in a class must be private.
- (f) In a class you can have more than one constructor with the same name.
- (g) If you declare a const object, it can only be used with const member functions.
- (h) When arguments are passed by value, the function works with the original arguments in the calling program.
- (i) Overloaded functions must have the same number and types of parameters.
- (j) Data items in a structure are public by default.

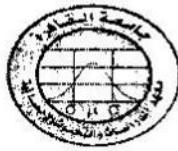
Q6) Write a program that uses a function called age_days() to satisfy the following requirements:

- The function takes four int values—for years, months, weeks, and days—as arguments, and returns the equivalent age in days (type long).
- With each call, the function displays a message telling how many times it has been called.: “I have been called 5 times”, for instance.
- The program repeatedly asks the user to enter a new age value represented in years, months, weeks and days, then calls the function, and displays the age measured by days. **(8 points)**

Q7) Create an employee class. The member data includes an int variable for storing the employee number and a float variable for storing the employee's salary. Define a set of member functions for allowing the user to enter and display the defined data members. Also, define a member function get_net_salary() for computing the net salary (salary - tax). This function computes the tax amount according to the following table. Write a main() function that allows the user to enter data for three employees then displays the employee number, salary and the net salary. **(8 points)**

Criteria	salary > =5000	5000>salary>=2000	salary<2000
Tax percentage	8%	4%	2%

Best wishes



ANSWER THE FOLLOWING QUESTIONS.

MAXIMUM DEGREE: 70 POINTS

- Q1)** Write a C++ program that accepts 100 integer numbers then sort the entered values in ascending order (from smallest to largest). Repeatedly, prompt the user to enter an integer value to search for. Finally, display the index of the first and last appearances of the entered number if it is found else display the message “the entered element is not found”. **(10 points)**

Q2. Write a recursive function *productOfMultiple(unsigned int x, unsigned int m)* that when invoked returns the summation of all multiples of *y* within the integer range $[x, 1]$. For instance, calling the function such as *productOfMultiple(5, 2)* returns the value 8, and *productOfMultiple(10, 3)* returns the value 162. **(6 points)**

(6 points)

Q3. Rewrite the following loop using both of *for* and *do-while* looping structures and Show the loop output. (6 points)

Q4) What is the output of executing the following code?

(18 points)

<pre>A #include <iostream.h> int main() { int x[4] = { 4,3,2 }; int* ptr; ptr=x; for(int j=0; j<4; j++) cout << "\n" << *ptr + j << "\t" << x[j]; return 0; }</pre>	<pre>B #include<iostream.h> int main() { int x=10; switch (x+2) { case 10: cout<<"hi"; break; case 11: cout<<"\n the ring"; break; default: cout<<"\n holla"; } return 0; }</pre>	<p>The result</p> <p>.....</p>	<p>The result</p> <p>.....</p>
<pre>C #include<iostream.h> void sp(float j, float* i, int & count) { static int m=0; float n; m++; count=m; n=j*2; j=*i*2; *i=n; } int main() { float x=10, y=5; int c; sp(x, &y, c); cout << "\n x , y and c are : " << x << "\t " << y << "\t" << c << endl; sp(x+2, &y, c); cout << "\n x , y and c are : " << x << "\t " << y << "\t" << c << endl; return 0; }</pre>	<pre>D #include <iostream.h> int main() { int x=1, y=3, z=1; bool b, c=0; b=x<y==++z>y; cout << "\n " << b; b=z<-1==x++>-y; cout << "\t" << b; c=27-30/2; cout << "\t" << c; x=++x+x--; cout << "\t" << x; return 0; }</pre>	<p>The result</p> <p>.....</p>	<p>The result</p> <p>.....</p>

Q5) State which of the following are true and which are false.

(10 points)

- (a) Data members in a class are public by default.
 - (b) If there are more initializers than the number of elements in the array, the array elements are automatically initialized starting from the first initializer and the rest of initializers will be ignored.
 - (c) When automatic variables are initialized, the initialization takes place only once with the first call for their function
 - (d) In a class definition, data or functions designated private are accessible only to member functions of that class.
 - (e) It is preferred to declare the data items in a structure data type to be private.
 - (f) In a class you can have more than one constructor with the same name and different parameters.
 - (g) If you declare a const object, it can only be used with const member functions.
 - (h) When arguments are passed by reference, the function works with the original arguments in the calling program.
 - (i) Overloaded functions must have the same return data type.
 - (j) An array can store many different values but from the same data type.

Q6) Write a program that achieves the following tasks: **Define** the structure *structTime* that includes **3** data items to store an integer time value for *hours*, *minutes*, and *seconds*. Then, define a function called *timeInSeconds()* that takes one parameter from type *structTime* and *returns the equivalent time* in seconds (type *long*). **With each call**, the function displays a message telling *how many times* it has been called: "I have been called 3 times", for instance. The program, **repeatedly**, asks the user to enter a new time value in hours, minutes, and seconds then calls the function *timeInSeconds()* and displays the value of that time in seconds. **(10 points)**

Q7) Create an employee class. The data members include the *employee number (int)* , the *tax percentage (float)* and the *employee's salary (float)*. Define a set of member functions to set and display the values of the defined data members. Then, define a member function *get_net_salary()* for computing the *net salary = salary - tax amount*. Also, define a member function *disp_income_level()* to display the income level of the employee according to the following table. Write a main() function that allows the user to enter data for three employees then displays the employee number, salary , the net salary and the income level. **(10 points)**

Criteria	net salary >=5000	5000>net salary>=2000	net salary<2000
Income level	“high”	“moderate”	“low”

Problem (1) Trace the following program **(10 points)**

```
#include <iostream>
#include <new>
using namespace std;
int main ()
{
int i,n;
int * p;
cout << "How many numbers would you like to
type? ";
cin >> i;
p= new (nothrow) int[i];
if (p == 0)
cout << "Error: memory could not be
allocated";
else
{ for (n=0; n<i; n++) {
cout << "Enter number: ";
cin >> p[n]; }
cout << "You have entered: ";
for (n=0; n<i; n++)
cout << p[n] << ", ";
delete[] p; }
return 0 ;}
```

Problem (2) what does this programs do? **(10 points)**

a) #include <iostream> **(5 points)**

```
using namespace std;
int main ()
{ int firstvalue, secondvalue;
int * mypointer;
mypointer = &firstvalue;
*mypointer = 10;
mypointer = &secondvalue;
*mypointer = 20;
cout << "firstvalue is " << firstvalue << endl;
cout << "secondvalue is " << secondvalue <<
endl;
return 0; }
```

```
using namespace std;
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;
for (int n=0; n<5; n++)
cout << numbers[n] << ", ";
return 0; }
```

Problem (3) **(10 points)**

Write a structure to the visitor information that contains fields for name, Email, mobile number, phone, ID. Then write a program to test it using in your program pointer to this structure and test it.

Problem (4) What does this program do **(10 points)**

```
#include <iostream>
using namespace std;
class CRectangle {
int *width, *height;
public:
CRectangle (int,int);
~CRectangle ();
int area () {return (*width * *height);}
};
CRectangle::CRectangle (int a, int b) {
width = new int;
height = new int;
*width = a;
*height = b;}
CRectangle::~CRectangle () {
delete width;
delete height;}
int main () {
CRectangle rect (3,4), rectb (5,6);
cout << "rect area: " << rect.area() <<
endl;
cout << "rectb area: " << rectb.area() <<
endl;
return 0; }
```

```
#include <iostream>
using namespace std;
int operate (int a, int b)
{ return (a*b);}
float operate (float a, float b)
{ return (a/b);}
int main ()
{ int x=5,y=2;
float n=5.0,m=2.0;
cout << operate (x,y);
cout << "\n";
cout << operate (n,m);
cout << "\n";
return 0; }
```

Problem (6)**(10 points)**

Write a program that prints out the larger of two integers entered from the keyboard. Use a function to do the actual comparison of the two numbers. Pass the two numbers to the function as arguments, and the function return the answer.

Problem (7) what is the output of this program (10 points)

a) Write a function that substitute each digit by another one from the following

S_box=[9,2,15,3,4,7,11,13,8,1,0,14,5,10,12,6]

b) Write a function that does the inverse of part a).

c) Inputs should be entered in the main program via the keyboard. Input should be tested for not be out of range from zero to fifteen. Test the two functions operation.



ANSWER THE FOLLOWING QUESTIONS.

MAXIMUM DEGREE: 70 POINTS

Q1) Write a program that uses a function called age_by_days() to satisfy the following requirements:

- The function takes one parameter from type structure age-struct that includes four int data members—for years, months, weeks, and days.
- The functions returns the equivalent age in days (type long).
- With each call, the function uses a static variable for displaying a message telling how many times it has been called.: “I have been called 5 times”, for instance.
- The program repeatedly asks the user to enter a new age value represented in years, months, weeks and days, then calls the function, and displays the age by days. **(8 points)**

Q2) Write a C++ program using two nested loops to draw the following

*

*

shape. All asterisks (*) should be printed by a single statement of the form cout << "*"; inside the loop body.

Q3) Write a recursive function factorial that calculates factorials where:

(5 points)

Factorial of $x = x * (x-1) * (x-2) * (x-3) * \dots * 3 * 2 * 1$

Q4) State which of the following are true and which are false.

(10 points)

- (a) A structure can store many different types of values.
- (b) If there are more initializers than the number of elements in the array, the remaining initializers are automatically ignored.
- (c) "\a " position the screen cursor to the beginning of the current line.
- (d) In a class, data members designated public are not accessible to member functions of that class.
- (e) All member functions in a class must be private.
- (f) If you declare a const object, it can only be used with const member functions.
- (g) When arguments are passed by reference, the function works with the original arguments in the calling program.
- (h) Overloaded functions must have the same number and types of parameters.
- (i) Data items in a structure are private by default.
- (j) The storage class of a variable means how long time the variable is alive.

Q3) What is the output of executing the following code?

(16 points)

A	<pre>#include <iostream.h> int main() { int x[4] = { 4,3, 2 }; int* ptr; ptr=x; for(int j=0; j<4; j++) cout << " " << *(x+j) * x[j] << endl; return 0; }</pre>	B	<pre>#include <iostream.h> int main() { int x =8; do { x++; if (x%3==0) continue; cout << "\n " << x; } while(x<20); return 0; }</pre>
C	<pre>#include<iostream.h> void sp(float* j, float* i, int* count) { static int c=0; float n; c++; *count=c; n=*j*2; *j=*i*2; *i=n; } int main() { float x=100, y=200; char ch; int c; sp(&x, &y, &c); cout << "\n x and y become : " << x << " and " << y; cout << "\n the value of c is: " << c; sp(&x, &y, &c); cout << "\n x and y become : " << x << " and " << y; cout << "\n the value of c is: " << c; return 0; }</pre>	D	<pre>#include <iostream.h> int main() { int x=3, y=-2, z=1; bool b, c=0; b=x<y==++z>y; cout << "\n " << b; b=z<-1==x++>-y; cout << "\t " << b; c= x/y*10; cout << "\t " << c; x=++x+x--; cout << "\t " << x; return 0; }</pre>
E	<pre>#include<iostream.h> int main() { int x=2; switch (x) { case 1: cout << "a"; case 2: cout << "b"; case 3: cout << "c"; } return 0; }</pre>	F	<pre>#include <iostream.h> #include <cstring> int main() { char str1[] = "int x = 10"; int MAX = strlen(str1); char str2[100]; for(int j=1; j<=MAX; j++) str2[MAX-j] = str1[j-1]; str2[MAX]='\0'; cout << str2 << endl; return 0; }</pre>
G	<pre>#include <iostream.h> class emp { private: static int cnt; int id; public: emp() { cnt+=10; id=cnt; } int getcnt() { return cnt; } int getId() { return id; } }; int emp::cnt = 0; int main() { int x; emp f1, f2; cout << "\n " << f1.getcnt() << "\t" << f1.getId(); emp f3, f4; cout << "\n " << f3.getcnt() << "\t" << f4.getId(); return 0; }</pre>	H	<pre>#include <iostream.h> void repchar() { for(int j=0; j<4; j++) cout << '*'; cout << endl; } void repchar(char ch) { for(int j=0; j<4; j++) cout << ch; cout << endl; } int repchar(char ch, int n) { for(int j=0; j<n; j++) cout << ch; cout << endl; return j+1; } int main() { repchar(); repchar('+', 7); repchar('='); return 0; }</pre>

and store them in a two-dimensional array, then computes and stores the sum of the grades for each student in the defined array. Finally attempt to display each student has a total grade greater than the average total grade and also display the student alphabetic grade according to the following table:

(8 points)

Total grade (tg)	$tg < 60$	$60 \leq tg < 75$	$75 \leq tg < 85$	$85 \leq tg < 95$	else
Alphabetic grade	Failed	Pass	Good	V Good	Excellent

Q7) Write two nested **for** loop that calculate and print the sum of the selected items.

(assume that: array X is declared)

(5 points)

X	0	1	2	3
0				
1				
2				
3				
4				

Q8) Rewrite the following code using **switch** statement.

(5 points)

```
int m, n, x;
cin>>m>>n>>x;
if(x==1||x==2)
    cout <<m-n;
else if(x==3)
    cout <<m+n;
else
    cout <<m*n;
```

Q9) Create an employee class. The member data includes an int variable for storing the employee number, a float variable for storing the employee's salary, a float variable for store the tax percentage. Define a set of member functions for allowing the user to enter and display the defined data members. Also, define a member function `get_net_salary()` for computing the monthly net salary . Then define a member function `get_annual_income()` that computes the annual income for the employee. Write a `main()` function that repeatedly asks the user to enter data for an employee then displays the employee number, salary , the net salary and the annual income.

(8 points)

Best wishes

Dr. Ahmed Gadallah