|  |  |  |
| --- | --- | --- |
| American University of SharjahCollege of Engineering Dept of Computer Science & Engg  P. O. Box 26666  Sharjah, UAE |  | Instructors: Dr. Rana Ahmed **Lab Instructor:** Eng. Sameer Alawnah  **Office: EB2-101**  **Phone**: +97165152974  **e-mail**: salawnah@aus.edu  **Semester**: Summer 2016 |

**CMP 120L– Introduction to Computer Science I Lab**

**Lab 13**

**Exercise 1:**

Write a main program that asks the user to enter the size, N, of a dynamic array that stores the GPA obtained by students as double numbers.

Your program should perform the following steps in this sequence:

1. Create the dynamic array (of type double) with the user-specified size, N. Check whether the dynamic array allocation was successful or not. If unsuccessful, exit the program and no further steps should be executed.
2. Print the starting address of the dynamically allocated array created in (1).
3. Create a loop that allows the user to enter a student’s GPA into corresponding array element, one by one.
4. Compute the average GPA, and output the result.
5. Deallocate (delete) memory allocated for your dynamic array before the end of main().

Test the working of your program with N=5.

#include <iostream>

using namespace std;

void main()

{

int N;

double a = 0, average;

double \*p;

cout << "please enter the number of students: ";

cin >> N;

p = new double[N];

cout << "Enter the students GPA: " << endl;

if (p == NULL)

{

cout << "Error insufficient memory" << endl;

exit(0);

}

for (int i = 0; i < N; i++)

{

cin >> p[i];

}

for (int i = 0; i < N; i++)

{

a = (p[i]) + a;

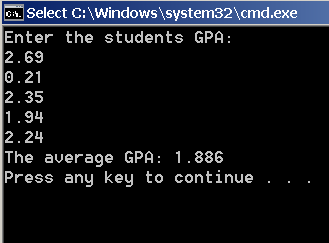
}

average = a / N;

cout << "The average GPA: " << average << endl;

delete[]p;

}



**Exercise 2:**

Write a C++ function **int f1 (int\* p, int N);** that accepts an integer array of size **N**. The array is passed as a pointer to its beginning. The starting address of the array is given by pointer **p**. The function returns the smallest element present in the array. For example, if in the main(), we have int x[4]= {11,3,20,9}; then the call f1(x,4) returns 3.

Test the above function by writing an appropriate main() program. All printouts should be done in main().

#include <iostream>

using namespace std;

int f1(int\* p, int N);

void main()

{

int \*p;

int min;

int x[4] = { 11, 3, 20, 9 };

min = f1(x, 4);

cout << "The value of the smallest in the array: " << min << endl;

}

int f1(int\* p, int N)

{

int min = p[0];

for (int i = 0; i < N; i++)

{

if (p[i]<min)

{

min = p[i];

}

}

return (min);

}



**Exercise 3:**

Write a C++ function **void copy (int\* p, int\* dest, int N);** that accepts an integer array of size **N,** with the starting address of the array given by pointer **p**. The function copies the elements of array pointed by **p** (i.e., with starting address **p**) into the array pointed by pointer **dest (i.e.,** with starting address **dest)** in the same order. The copying process is done element by element. For example, if in main(), we have int x[5]= {10,20,30,40,1}, y[5]; int \*px=x; int \*py=y;

then after the function call copy(px, py, 5), we should have the y array as {10,20,30,40,1} in the main().

Test the above function by writing an appropriate main() program. All printouts should be done in main().

#include <iostream>

using namespace std;

void copy(int\* p, int\* dest, int N);

void main()

{

int x[5] = { 10, 20, 30, 40, 1 };

int y[5];

copy(x, y, 5);

cout << "Array y is: " << endl;

for (int i = 0; i < 5; i++)

{

cout<< y[i] << endl;

}

}

void copy(int\* p, int\* dest, int N)

{

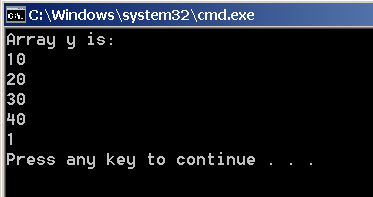
for (int i = 0; i < 5; i++)

{

dest[i] = p[i];

}

}



**Exercise 4:**

A string is a special character array with the property that the last element in the string is a special character (‘\0’), also known as a NULL (or terminating) character. One way to declare and initialize string is to use statements such as : char z[]= “Hello”;

Write a C++ function **int count(char\* p);** that accepts a string with starting address **p**, and returns the number of lowercase vowels present in the string. The vowels are characters ‘a’, ‘e’, ‘i’, ‘o’, and ‘u’. For example, if we have char char z[]= “Hello”; char t[]= “December”; in main(), then the call count(z) should return 2, while the call count(t) should return 3.

Please note that the length of the string is not passed as one of the parameter in function count(), as the last terminating character (‘\0’) will indicate the end of the string.

Test the above function by writing an appropriate main() program. All printouts should be done in main().

#include <iostream>

using namespace std;

int count(char\* p);

void main()

{

int x;

char z[] = { 'H', 'e', 'l', 'l', 'o', '\0' };

x = count(z);

cout << "The number of lowercase vowels is: " << x << endl;

char t[] = { 'D', 'e', 'c', 'e', 'm', 'b', 'e', 'r', '\0' };

x = count(t);

cout << "The number of lowercase vowels is: " << x << endl;

}

int count(char\* p)

{

int i = 0;

while ((\*p) != '\0')

{

if (((\*p) == 'a') || ((\*p) == 'e') || ((\*p) == 'i') || ((\*p) == 'o') || ((\*p) == 'u'))

{

i++;

}

p++;

}

return (i);

}

