**Homework 9**

IM/2019/094

H.P.S. Gandhira

**1.Declare two (type double) pointer variables named d\_var and d\_array. Write C++ statements to dynamically create a double‐precision floating-point variable and store its address in d\_var. Also dynamically create an array of 10 double‐precision floating‐point values and store its address in d\_array:**

double \*d\_var ;

double \*d\_array ;

d\_var = new double;

d\_array = new double[10];

**2.Write C++ statements to input a value for d\_var (i.e., a value that d\_var points to) from the  console and then display it**

#include <iostream>

using namespace std;

int main(){

double \*d\_var ,\*d\_array;

d\_var=new double;

d\_array=new double[10];

cout<<"enter the value into the variable : ";

cin>> \*d\_var;

cout <<\*d\_var<<endl;

}

**3.Write C++ statements to initialize the 10 double values in the dynamically allocated array to  1.0. Now write C++ statements to de- allocate the memory (i.e. using the delete operator) pointed by d\_var and d\_array**

#include <iostream>

using namespace std;

int main(){

double \*d\_var, \*d\_array;

d\_var=new double;

d\_array=new double[10];

for(int i=0; i<10; i++){

d\_array[i]= 1.0;

cout<<" "<< d\_array[i];

}

delete [] d\_array;

delete [] d\_var;

}

**4.Write a program in C to find the largest element using Dynamic Memory Allocation**

#include <iostream>

using namespace std;

void findLargest(int\* arr, int N){

for (int i = 1; i < N; i++) {

if (\*arr < \*(arr + i)) {

\*arr = \*(arr + i);

}

}

cout << \*arr;

}

int main(){

int N ;

int\* arr;

cout << " Input number of elements :";

cin >> N;

arr = new int[N];

if (arr == NULL) {

cout << "No memory allocated";

}

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

cout << " Input array "<<i+1<<" value: ";

cin >> arr[i];

}

cout << " Largest value is : ";

findLargest(arr, N);

return 0;}

**5.A user‐defined structure named Timer has two integer parameters to initialize hour and minute members. Write a single C++ statement to create an object of the Timer structure using dynamic memory allocation and assign it to a pointer variable named timePtr. Assign values of 10 and 20 for hour and minute respectively.**

#include <iostream>

#include <stdlib.h>

using namespace std;

struct timer{

int hr;

int min; };

int main(){

timer \*timeptr = new timer{10,20};

return 0;

}

**6.Letter Frequency  Write a function that will take a string and return a count of each letter in the string. For  example, "my dog ate my homework" contains 3 m's, 3 o's, 2 e's, 2 y's and one each of d, g,  a, t, h, w, r and k.  Your function should take a single string argument and return a dynamically allocated array  of 26 integers representing the count of each of the letters a ‐ z respectively. Your function  should be case insensitive, i.e., count ‘A’ and ‘a’ as the occurrence of the letter ‘a’. Do not  count non‐ letter characters (i.e., spaces, punctuation, digits, etc.) Write a program that will take a string from the user using cin, call your letter frequency function and print out the frequency of each letter in the string. Do not list letters that do not occur at least once**

#include <iostream>

#include <cstring>

#include <string>

using namespace std;

int \* letter\_f(char s[]);

int main(){

int \*frq\_ltrs;

char \*s = new char[100];

frq\_ltrs = new int[26];

cout << " Enter a string here : ";

cin.getline(s, ' ');

frq\_ltrs = letter\_f(s);

cout << " Letter frequency :"<< endl;

for(int i = 0; i<26; i++){

if(frq\_ltrs[i] != 0)

cout << " "<< static\_cast<char>(i + 'a') << " " << frq\_ltrs[i] << endl;

}

system("pause");

return 0;

}

int \*letter\_f(char s[]){

int \*occ\_arr;

occ\_arr = new int[26];

int count;

for(int i = 0; i <26; i++){

count = 0;

for(int j=0; j <strlen(s); j++){

if (int('a') + i ==int(tolower(s[j])))

count++;

}

occ\_arr[i] = count;

}

return occ\_arr;

}