Problem 1: Write the program that reads student scores, gets the best score, and then assigns grades based on the following scheme:

Grade is A if score is >= best-10;

Grade is B if score is >=best-20;

Grade is C if score is >=best-30;

Grade is D if score is >=best-40;

Grade is F otherwise.

The program prompts the user to enter the total number of students, then prompts the user to enter all of the scores, and concludes by displaying the grades.

Program code

#include <iostream>

using namespace std;

int main()

{

int score[100];

int n,i,max;

max=0;

cout<<"Enter the number of students: ";

cin>>n;

cout<<endl;

cout<<"Enter 4 scores: ";

for (i=0;i<=(n-1);i++)

{

cin>>score[i];

if (score[i]>=max) max=score[i];

}

for (i=0;i<=(n-1);i++)

{

if (score[i]>=(max-10)) cout<<"Student "<<i<<" score is "<<score[i]<<" and grade is A"<<endl;

else if (score[i]>=(max-20)) cout<<"Student "<<i<<" score is "<<score[i]<<" and grade is B"<<endl;

else if (score[i]>=(max-30)) cout<<"Student "<<i<<" score is "<<score[i]<<" and grade is C"<<endl;

else if (score[i]>=(max-40)) cout<<"Student "<<i<<" score is "<<score[i]<<" and grade is D"<<endl;

else cout<<"Student "<<i<<" score is "<<score[i]<<" and grade is F"<<endl;

}

return 0;

}

Program analysis

1. using a loop statement to read the number and store them in a array.
2. Using var max to get to best score at the same time
3. Using if statement to check the final grade;

Program result

Problem 2:

Write two overloaded functions that return the average of an array wit the following headers:

int average(const int array[], int size);

double average(const double array[], int size);

Write a test program that prompts the user to enter 10 double values, invoke this function, and display the average value.

Program code

#include <iostream>

using namespace std;

int average(const int array[], int size);

double average(const double array[], int size);

int main()

{

double number[10];

int i;

cout<<"Enter 10 number: ";

for (i=0;i<=9;i++)

{

cin>>number[i];

}

cout<<"The average is "<<average(number,10)<<endl;

return 0;

}

int average(const int array[], int size)

{

int a;

int sum=0;

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

{

sum=sum+array[i];

}

a=sum/size;

return a;

}

double average(const double array[], int size)

{

double a;

double sum=0;

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

{

sum=sum+array[i];

}

a=sum/size;

return a;

}

Program analysis

using overloaded function to distinguish different variable type.

Program result

Problem 3:

Write the following function that returns true if the list is already sorted in increasing order:

bool isSorted(const int list[], int size)

Write a test program that prompts the user to enter a list and displays whether the list is sorted or not. Here is a sample run. Note that the first number in the input indicates that the number of the elements in the list. This number is not part of the list. Assume the maximum list size is 80

Program code

#include <iostream>

using namespace std;

bool isSorted(const int list[], int size);

int main()

{

int n,i;

int list[80];

cout<<"Enter list: ";

cin>>n;

for (i=0;i<=(n-1);i++)

cin>>list[i];

if (isSorted(list,n)==true )

cout<<"The list is sorted"<<endl;

else cout<<"The list is not sorted"<<endl;

return 0;

}

bool isSorted(const int list[], int size)

{

int i;

bool check;

check=true;

for (i=1;i<=size;i++)

{

if (list[i]<list[i-1])

check=false;

}

return check;

}

Program analysis

In the function isSorted, use a loop statement to check that whether list[i] is smaller than or equal to list[i-1]

Program result

Problem 4:

Write the following function that partitions the list using the first element, called a pivot:

int partition(int list[], int size)

After the partition, the elements in the list are rearranged so that all the elements before the pivot are less than or equal to the pivot and the elements after the pivot are greater than the pivot. The function also returns the index where the pivot is located in the new list. For example, suppose the list is {5,2,9,3,6,8}. After the partition, the list becomes {3,2,5,9,6,8}. Implement the function in a way that takes size number of comparisons.

Write a test program that prompts the user to enter a list and displays the list after the partition. Here is a sample run. Note the first number in the input indicates that the number of elements in the list. This number is not part of the list. Assume that the maximum list size is 80.

Program code

#include <iostream>

using namespace std;

int partition(int list[], int size);

int main()

{

int n,i;

int list[80];

cout<<"Enter list: ";

cin>>n;

for (i=0;i<=(n-1);i++)

cin>>list[i];

cout<<"The location of the pivot is "<<partition(list,n)<<endl;

for (i=0;i<=(n-1);i++)

{

cout<<list[i]<<" ";

}

cout<<endl;

return 0;

}

int partition(int list[], int size)

{

int pivot=list[0];

int i,location;

location=1;

int array[80];

for (i=1;i<=(size-1);i++)

{

if (list[i]<pivot)

location++;

}//count the number before the pivot

int front,behind;

front=0;behind=0;

array[location-1]=pivot;

for (i=1;i<=(size-1);i++)

{

if (list[i]>=pivot)

{

array[size-1-behind]=list[i];

behind++;

}

else

{

array[front]=list[i];

front++;

}

}

for (i=0;i<=(size-1);i++)

{

list[i]=array[i];

}

return location;

}

Program analysis

1. Comparing the every number to find the location of the pivot
2. Using a new array to transfer the number
3. Count the number from the head and from the end at the same time, to change the location of each number

Program result

Problem 5:

Write a function that counts the occurrence of each letter in the string using following header:

void count(const char s[], int counts[])

where counts is an array of 26 integers. counts[0], counts[1],…, and counts[25] count the occurrence of a,b,…,z respectively. Letters are not case-sensitive, i.e., letter A and a are counted the same as a.

Write a test program that reads a string, invokes the count function, and displays the non-zero counts

Program code

#include<iostream>

#include <string>

using namespace std;

void count(const char s[], int counts[]);

int main()

{

int i;

char s[80];

int counts[26];

gets(s);

count(s, counts);

return 0;

}

void count(const char s[], int counts[])

{

int i,j;

for(i=0;i<=25;i++)

{

counts[i]=0;

} // 把counts都变成0

char character[26];

for (i=0;i<=25;i++)

{

character[i]='A'+i;

}

int ascll[26];

for(i=65;i<=90;i++)

{

ascll[i-65]=(i-65);

}//把ascll 以此变成0,1,2，....25

for(i=0;i<=80;i++)

{

for (j=0;j<=25;j++)

{

if ((s[i]-'A')==j || (s[i]-'A')==(j+32))

counts[j]++;//得出[s]的ascll码，然后做比较

}

}

for(i=0;i<=25;i++)

{

if (counts[i]==0);

else cout<<"The number of the character "<<character[i]<<" is "<<counts[i]<<endl;//如果那个个数不为0，输出那个个数

}

}

Program analysis

1. using ASCLL code to check which character the char array refer to .
2. create several array to store the sequential letter
3. using two loop statement to count the number of occurrence.

Program result