Name – Ankesh Kumar
NU ID - 002208893

Homework 1

Problem 1 –

Part 1. Pass-by-pointer

Approach for solving the problem:

The question asks for implementation of function which would swap two integers by pointer method.

Coming to the basics, the int x and *p holds the value while &x and p holds the address in memory where that value is stored.

1. We initialise two integers, x = 10 and y = 20 and pass the address's of these variables in function as - swapP(&x, &y);
2. Inside function, it takes int* a, int* b which refers to the value which holds.
3. A temp variable is initialized which stores *a which is dereferencing to the value from address.
4. Then we move the *b value to *a and then place the temp value back to *a.

Summary :

In the above mentioned steps, we by passing the address, and inside function, de-referencing the pointers, we swap the value.

When cout / return statement will happen, the same addresses/ variables will have the swapped values.

Part 2. Pass-by-reference

Approach for solving the problem :

Opposite to above mentioned, pass-by-reference means we pass the variable parameter into the function and refer to the address and make changes. If we change the value of variable in function, same would be reflected in the variable as value of the variable changes.

1. Initialize two variables a =3, b= 4.
2. Next, pass the variables to function - swapR(a,b); and while initializing the function, we mention - void swapR(int& numRef1, int& numRef2)
3. Now, performing the swap technique i.e. creating an temp variable and putting value of numRef1 and in numRef1 we put the value numRef2.
4. Finally, we set numRef2 from the temp variable.

Summary :

Here we pass the variable in function, and in function, instead of calling the variable, it calls the address. Then we make the required transactions and as the changes are made referring to the address, the value which we change, when called from main function is reflected.

EECE7205 – Fundamental of Computer Engineering

Name – Ankesh Kumar
NU ID - 002208893

Execution screenshot :

HW1 (Global Scope) swapP(in

```
7
8   //Function signature for the functions used in program
9   void swapP(int* numRef1, int* numRef2);
10  void swapR(int& numRef1, int& numRef2);
11
12  //Main function
13  int main() {
14      int a{ 3 };
15      int b{ 4 };
16      cout << "Before swapping values fo
17      swapR(a, b); //calling the functio
18      cout << "After swapping, the value
19
20      //New part - using Pass-by-Pointer
21      int x{ 10 };
22      int y{ 20 };
23      cout << "Before swapping values fo
24      swapP(&x, &y); //We pass the addre
25      cout << "After swapping, the value
26  };
27
28  //Function for Pass-by-Reference
29  void swapR(int& numRef1, int& numRef2)
30      int temp = numRef1; //Declaring ne
31      numRef1 = numRef2;
32      numRef2 = temp; //The values at th
33  };
34
35  //Function for Pass-by-Pointer
36  void swapP(int* numRef1, int* numRef2)
37      int temp = *numRef1; //Declaring n
38      *numRef1 = *numRef2; //Changing th
39      *numRef2 = temp;
40  }
41
```

90 %

Error List Output Data Tools Operations Web Publish Activ

Microsoft Visual Studio Debu

```
Before swapping values for a : 3 and b : 4
After swapping, the values for a : 4 and b : 3
Before swapping values for x : 10 and y : 20
After swapping, the values for x : 20 and y : 10

N:\Cprograms\VSPrograms\HW1\Debug\HW1.exe (process 26348)
To automatically close the console when debugging stops, e
le when debugging stops.
Press any key to close this window . . .
```

Name – Ankesh Kumar
NU ID - 002208893

Problem 2 –

Approach for solving the problem :

Here we have a main() statement and er need to develop a function **MinFirst().**

Breaking down the steps –

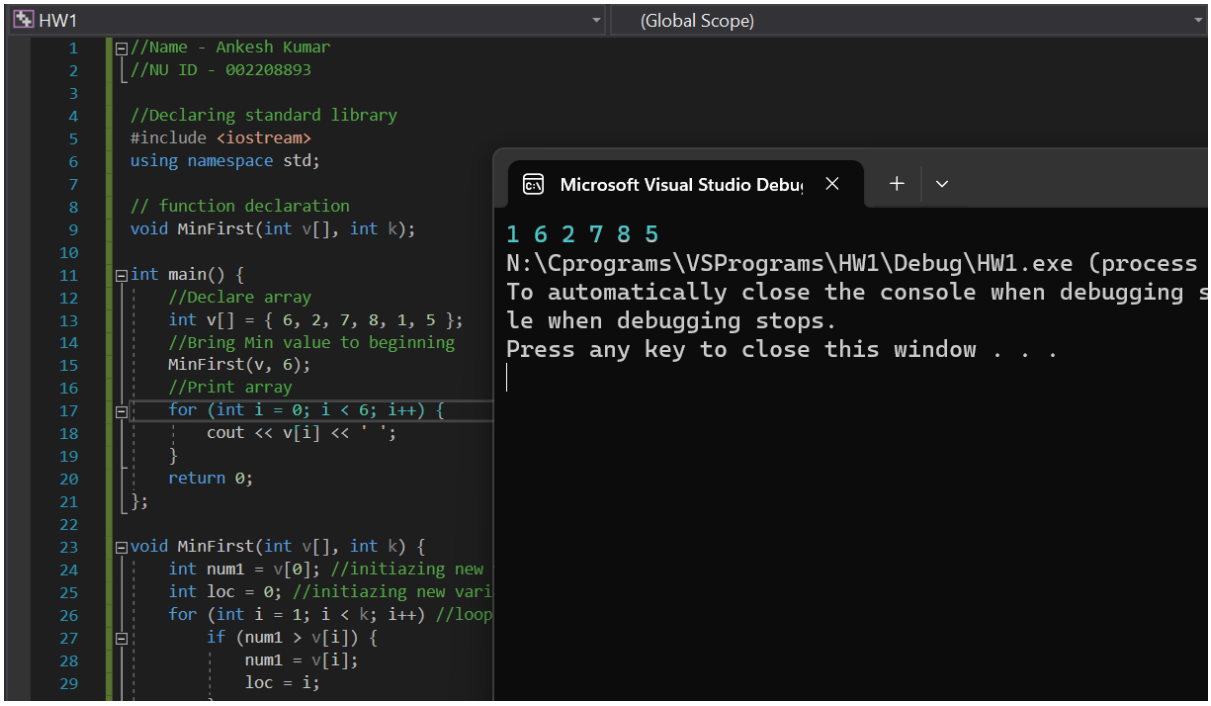
1. We initialise two variables num1 and loc which would store the Minimum value in array and the corresponding location.
2. In next step, we execute the loop through array and get the Minimum value and Location.
3. Next we initialise *numToreplace* variable and place the first element of the Array(i.e. v[0]) to it.
4. Then, we start a new loop and here we keep shifting the variable to one location further. Also, we have included an if-else statement, which works like, if there is same location of minimum value, after that shift is skipped and the loop breaks.
5. Next, we make the first element of loop equal to the minimum value.

Summary :

By this method, the loop executes till we find the location of minimum value and then breaks out of the loop.

This method gives the result as expected.

Execution screenshot :



Name – Ankesh Kumar
NU ID - 002208893

Problem 3 –

Approach for solving the problem :

The code has following aspects while creation

1. The question asks to create functions which would solve following aspects –
2. ***Struct*** for declaration ***Students*** to get the student data (LastName, Age and Grade)
   It is a struct statement where we declare the Structure and datatype.
3. ***GetRecord*** - Input method for the Struct.
   It is created to get input value for the each Student as per the number / size of class entered by user.
4. ***GetMedian*** – function to calculate median value of Ages
   a. Here, we first do the code the method on how to do insertion sort as Median calculation would require sorted array.
   b. In this, we follow the concept like on sorted array of Ages,
      i. if the size is even then Median is average of two middle values.
      ii. If the size is odd, then middle value of the list is Median.

   It is implemented using a if-else statement.

5. ***GetAverage*** - function to calculate Average value of grades
   a. We loop through all the Graade entity object in struct Student and sum them all.
   b. Next, we divide the sum of Grades and divide it by size

6. ***Get calculate values*** – Here we use the same method similar to ques 2, to get the top two values and print the data of Highest Two grade and Lowest two grade.
   a. For the first part – Getting highest Two grade
      i. Here we initialise two variable a, b as -1, considering that lowest grade would be 0.
      ii. Then, we use to find the minimum value by looping it through and recording the index/ location of the highest number.
      iii. Using that index, print the top two highest value.
   b. For the second part – Getting Lowest Two grade
      i. We just reverse the sign and here we compare one by one values from the loop from previous versus current.
      ii. And we collect the index of that lowest variable.
      iii. Using that location, we perint the lowest two grades

7. Final ***main()*** function where we ask size of class from teacher and accordingly perform Dynamic memory allocation and then delete the Struct at the program end.

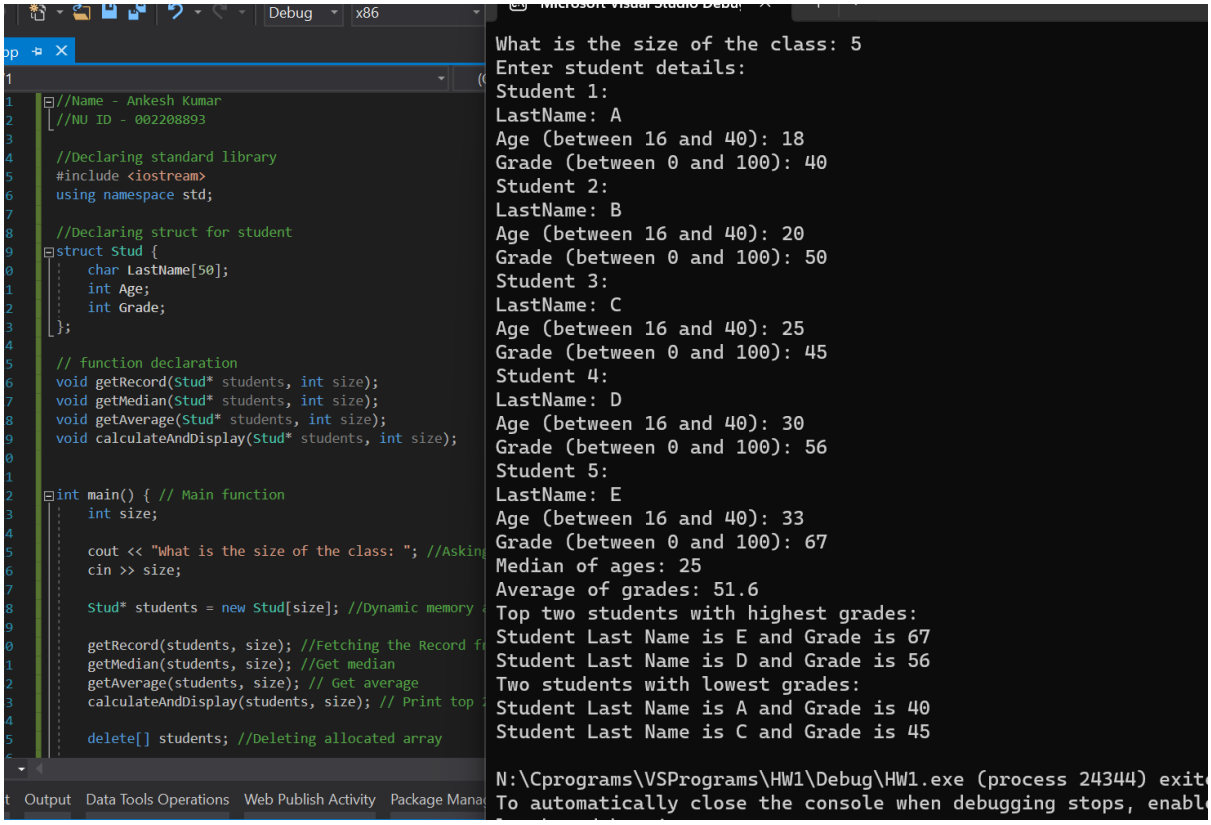
Summary :

Formed the C++ program with all the asked functionality as well as considering Dynamic memory allocation.

Name – Ankesh Kumar
NU ID - 002208893

The code also utilises InsertionSort to sort the array for getting median which is an extra part as sorting was needed.

Execution screenshot :



```
//Name - Ankesh Kumar
//NU ID - 002208893

//Declaring standard library
#include <iostream>
using namespace std;

//Declaring struct for student
struct Stud {
    char LastName[50];
    int Age;
    int Grade;
};

// function declaration
void getRecord(Stud* students, int size);
void getMedian(Stud* students, int size);
void getAverage(Stud* students, int size);
void calculateAndDisplay(Stud* students, int size);


int main() { // Main function
    int size;

    cout << "What is the size of the class: "; //Asking
    cin >> size;

    Stud* students = new Stud[size]; //Dynamic memory

    getRecord(students, size); //Fetching the Record f
    getMedian(students, size); //Get median
    getAverage(students, size); // Get average
    calculateAndDisplay(students, size); // Print top

    delete[] students; //Deleting allocated array
```

```
What is the size of the class: 5
Enter student details:
Student 1:
LastName: A
Age (between 16 and 40): 18
Grade (between 0 and 100): 40
Student 2:
LastName: B
Age (between 16 and 40): 20
Grade (between 0 and 100): 50
Student 3:
LastName: C
Age (between 16 and 40): 25
Grade (between 0 and 100): 45
Student 4:
LastName: D
Age (between 16 and 40): 30
Grade (between 0 and 100): 56
Student 5:
LastName: E
Age (between 16 and 40): 33
Grade (between 0 and 100): 67
Median of ages: 25
Average of grades: 51.6
Top two students with highest grades:
Student Last Name is E and Grade is 67
Student Last Name is D and Grade is 56
Two students with lowest grades:
Student Last Name is A and Grade is 40
Student Last Name is C and Grade is 45

N:\Cprograms\VSPrograms\HW1\Debug\HW1.exe (process 24344) exit
To automatically close the console when debugging stops, enabl
```