==========

Roll Number: SYCOC303 Division: C

PRN Number: 122B2B303 Batch: C4

Name: VINAYAK MADAN SHETE

==========

Problem Statement:

- **Write a C++ program to store the monthly salary of an employee in an array. Sort array of numbers in ascending order using Merge sort and Display details of top five employees with the highest salary.**

==========

INPUT:

```
/*
 *     ====================================
 *            Program Name: MergeSort.cpp
 *     Created on: December 22, 2022
 *       Author: Vinayak Shete
 *  ====================================
 */

#include <iostream>
#include <string.h>
using namespace std;

struct EmpDet
{
	int emp_id;
	string emp_name;
	int emp_salary;
};

//This method merges the partitions after sorting them
void Merge(struct EmpDet *a, int low, int high, int mid)
{
```

```
    int i, j, k;
    struct EmpDet temp[high-low+1];
    i = low;
    k = 0;
    j = mid + 1;

//This loop sorts and merges the elements
    while (i <= mid && j <= high)
    {
        if (a[i].emp_salary < a[j].emp_salary)
        {
            temp[k] = a[i];
            k++;
            i++;
        }
        else
        {
            temp[k] = a[j];
            k++;
            j++;
        }
    }

    //J exceeded size and elements after i are left to add
    //therefore adding all the elements as it is to resultant array
    while (i <= mid)
    {
        temp[k] = a[i];
        k++;
        i++;
    }

    //I exceeded size and elements after j are left to add
    //therefore adding all the elements as it is to resultant array
    while (j <= high)
    {
        temp[k] = a[j];
```

```
			k++;
			j++;
		}

		//replacing sorted array from temporary to original
		for (i = low; i <= high; i++)
		{
			a[i] = temp[i-low];
		}
}

 //this method divides the array and calls merge to sort & merge the array elements
void MergeSort(struct EmpDet *a, int low, int high)
{
		int mid;
		if (low < high)
		{
			mid=(low+high)/2;
			MergeSort(a, low, mid);
			MergeSort(a, mid+1, high);

			Merge(a, low, high, mid);
		}
}

//function for displaying employee details
void display(struct EmpDet *a,int size)
{
		cout<<"\n============EMPLOYEE DETAILS==============";
		cout<<"\nID\t\tName\t\tSalary\n";
		for(int i=0;i<size;i++)
	{
		 cout<<a[i].emp_id<<"\t\t"<<a[i].emp_name<<"\t\t"<<a[i].emp_salary;
		 cout<<"\n";
	}
}
```

```
//displaying top 5 employees with highest salary
void top5(struct EmpDet *a, int size)
{
	cout<<"\n====TOP 5 EMPLOYEE DETAILS WITH HIGHEST SALARY====";
	cout<<"\nID\t\tName\t\tSalary\n";
	int z=0;
	for(int i=size-1;z<5;i--)
	{
		cout<<a[i].emp_id<<"\t\t"<<a[i].emp_name<<"\t\t"<<a[i].emp_salary;
	 cout<<"\n";
	 z++;
	}
}


int main()
{
	int n,i,id,salary,ch,doch;
	string name;
	cout<<"============WELCOME===========";
	cout<<"\nCreating Employee Details Table=>>";
	cout<<"\nHow many employees you want to add?-->";
    cin>>n;
    struct EmpDet empArr[n];
    cout<<"\nEnter "<<n<<" Employees Details:\n";
    for(i=0;i<n;i++)
    {
	cout<<"\n=================";
	cout<<"\nEnter details for employee "<<i+1<<"-->";
	cout<<"\nEnter employee id:";
	cin>>id;
	cout<<"\nEnter employee name:";
	cin>>name;
	cout<<"\nEnter employee salary:";
	cin>>salary;
	empArr[i].emp_id=id;
	empArr[i].emp_name=name;
```

```
        empArr[i].emp_salary=salary;
        cout<<"\n==================";
    }
    do
    {
        cout<<"\n1.Display\n2.Sort using Merge Sort\n3.Display top 5 employees with
highest salary\n4.EXIT";
            cout<<"\nEnter your choice:";
            cin>>ch;
            switch(ch)
            {
                case 1:
                    display(empArr,n);
                    break;

                case 2:
                    MergeSort(empArr,0,n-1);
                    cout<<"\nThe employee details have been sorted according to
their Salary!==>";
                    display(empArr,n);
                    break;

                case 3:
                    top5(empArr,n);
                    break;

                case 4:
                    goto exit;
                    break;

                default:
                    cout<<"\nPlease enter correct choice!";
            }
            cout<<"\n===\nDo you want to continue?[1 for YES || 0 for NO]-->";
            cin>>doch;
        }while(doch==1);
        exit:
```

```
            cout<<"\n=====THANK YOU==========";
   return 0;
}
```

==========

OUTPUT:

```
===========WELCOME==========
Creating Employee Details Table=>>
How many employees you want to add?-->7

Enter 7 Employees Details:

=================
Enter details for employee 1-->
Enter employee id:101

Enter employee name:ABC

Enter employee salary:50000

=================
=================
Enter details for employee 2-->
Enter employee id:102

Enter employee name:DEF

Enter employee salary:62000

=================
=================
Enter details for employee 3-->
Enter employee id:103

Enter employee name:GHI

Enter employee salary:45000

=================
=================
Enter details for employee 4-->
Enter employee id:104

Enter employee name:JKL

Enter employee salary:90000

=================
=================
```

```
Enter details for employee 4-->
Enter employee id:104

Enter employee name:JKL

Enter employee salary:90000

=================
=================
Enter details for employee 5-->
Enter employee id:105

Enter employee name:MNO

Enter employee salary:30000

=================
=================
Enter details for employee 6-->
Enter employee id:106

Enter employee name:PQR

Enter employee salary:42000

=================
=================
Enter details for employee 7-->
Enter employee id:107

Enter employee name:STU

Enter employee salary:100000

=================
```

```
==================
1.Display
2.Sort using Merge Sort
3.Display top 5 employees with highest salary
4.EXIT
Enter your choice:
```

Displaying before Sorting:

```
==================
1.Display
2.Sort using Merge Sort
3.Display top 5 employees with highest salary
4.EXIT
Enter your choice:1

============EMPLOYEE DETAILS==============
ID              Name            Salary
101             ABC             50000
102             DEF             62000
103             GHI             45000
104             JKL             90000
105             MNO             30000
106             PQR             42000
107             STU             100000

===
Do you want to continue?[1 for YES || 0 for NO]-->
```

Displaying after Sorting:

```
1.Display
2.Sort using Merge Sort
3.Display top 5 employees with highest salary
4.EXIT
Enter your choice:2

The employee details have been sorted according to their Salary!==>
============EMPLOYEE DETAILS==============
ID              Name            Salary
105             MNO             30000
106             PQR             42000
103             GHI             45000
101             ABC             50000
102             DEF             62000
104             JKL             90000
107             STU             100000

===
Do you want to continue?[1 for YES || 0 for NO]-->
```

Displaying top 5 with highest salaries:

```
1.Display
2.Sort using Merge Sort
3.Display top 5 employees with highest salary
4.EXIT
Enter your choice:3

====TOP 5 EMPLOYEE DETAILS WITH HIGHEST SALARY====
ID              Name            Salary
107             STU             100000
104             JKL             90000
102             DEF             62000
101             ABC             50000
103             GHI             45000

===
Do you want to continue?[1 for YES || 0 for NO]-->
```

==========