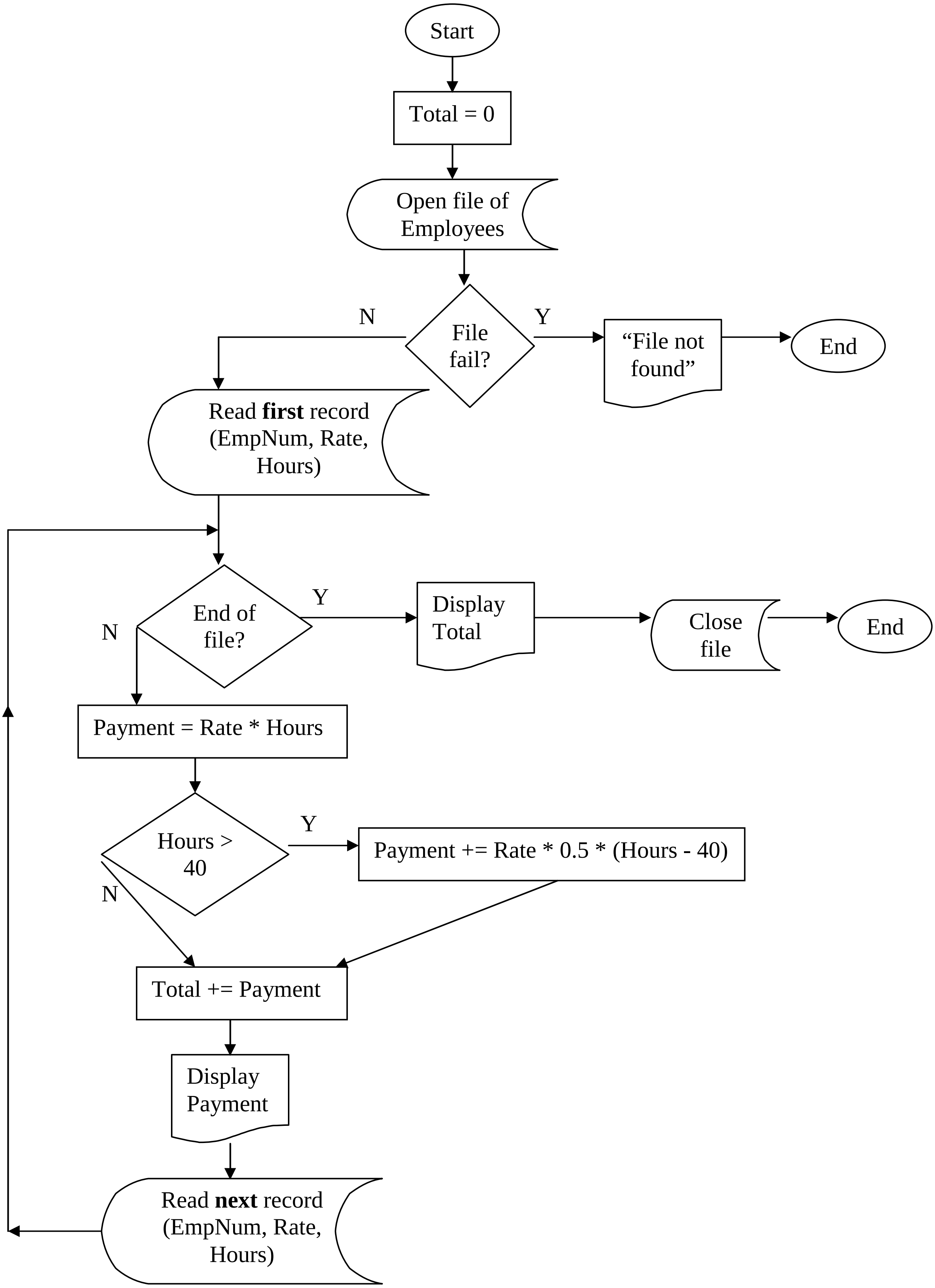
BIL105E - Introduction to Scientific and Engineering Computing
Final Exam - 30.5.2008 (KEYS)

Answer 1)



Answer 2)

```
#include <iostream.h>
#include <stdlib.h>

int main()
{
      int k, N;
      long int X;
      int which_number, length, max_length = -1;

      cout << "Enter upper limit: ";
      cin >> N;

      for (k=1; k <= N; k++)
      {
        cout << "Number= " << k << " , Ackermann Sequence= " << k << "  ";

        X = k;
        length = 1; // Count of Ackermann numbers

        // Generate all Ackermann numbers for X
        do
        {
           if (X % 2 == 0)
              X = X / 2;
           else
              X = 3 * X + 1;

           cout << X << "  ";

           length++;

        } while (X != 1);

        cout << "\n";

        // Compare with the longest length so far:
        if (length > max_length)
        {
           which_number = k;    // New number having longest sequence
           max_length = length; // New maximum sequence length
        }

      }
      cout << "\n\n";
      cout << "NUMBER HAVING THE LONGEST SEQENCE= " << which_number << "  ";
      cout << "WITH LENGTH= " << max_length << "\n\n";

      system("PAUSE");
      return 0;
}
```

Answer 3)

```
#include <iostream.h>
#include <stdlib.h>

void Display_Array(int X[], int NX)
{
  int i;

  for (i=0; i < NX; i++)
     cout << X[i] << "  ";
  cout << "\n\n";
}

// In the exam, only set functions were required.
//------------------------------------------------------------------------------
void Unification(int A[], int B[], int C[], int NA, int NB, int & NC)
{
  int i;
  int k = 0; // Counter index of C array

  for (i=0; i < NA; i++)
  {
     C[k] = A[i];
     k++;
  }

  for (i=0; i < NB; i++)
  {
     C[k] = B[i];
     k++;
  }

  NC = k; // Number of elements in C
}

//------------------------------------------------------------------------------
void Intersection(int A[], int B[], int C[], int NA, int NB, int & NC)
{
  int i, j;
  int k = 0; // Counter index of C array

  for (i=0; i < NA; i++)
  {
     for (j=0; j < NB; j++)
     {
        if (A[i] == B[j])
        {
          C[k] = A[i];
          k++;
        }
```

```
      }
   }

   NC = k; // Number of elements in C

}

//---------------------------------------------------------------------------------
void Difference(int A[], int B[], int C[], int NA, int NB, int & NC)
{
   int i, j;
   int k = 0; // Counter index of C array
   bool found;

   for (i=0; i < NA; i++)
   {
       found = false;
       for (j=0; j < NB; j++)
       {
           if (A[i] == B[j])
           {
              found = true;
              break; // Skip the rest of j loop
           }
       }

       if (!found)
       {
          C[k] = A[i];
          k++;
       }
   }

   NC = k; // Number of elements in C
}

//---------------------------------------------------------------------------------


int main()
{
   int NA, NB, NC;

   int A[] = {1,4,5,6,2};
   int B[] = {6,3,4,7};
   int C[NC];
   int i;

   NA = sizeof(A)/sizeof(int);
   NB = sizeof(B)/sizeof(int);
   NC = NA + NB;
```

```
  cout << "A = "; Display_Array(A, NA);
  cout << "B = "; Display_Array(B, NB);

  Unification(A, B, C, NA, NB, NC);
  cout << "Unification = "; Display_Array(C, NC);

  Intersection(A, B, C, NA, NB, NC);
  cout << "Intersection = "; Display_Array(C, NC);

  Difference(A, B, C, NA, NB, NC);
  cout << "Difference = "; Display_Array(C, NC);

  cout << "\n";
  system("PAUSE");
  return 0;
}
```

Answer 4)

```
#include <iostream.h>
#include <stdlib.h>
#include <iomanip.h> // setf, setw, etc.

struct SStudent
{
  char Name[10];
  int Age;
};
typedef struct SStudent TStudent;


int main()
{
 int N; // Number of students in the list
 int i, j;

 TStudent Student[] = {  {"Brown",    20},
                  {"Rovelli",  18},
                  {"Wong",     22},
                  {"Cramer",   18},
                  {"Gutier",   21}  };

 TStudent tmp;

 N = sizeof(Student) / sizeof(TStudent); // Find the count

 cout.setf(ios::left);
```

```
  cout << "ORIGINAL LIST:\n";
  for (i=0; i < N; i++)
     cout << setw(11) << Student[i].Name << " " << Student[i].Age << "\n";

  // Sort the array by name in ascending order:
  for (i=0; i < N-1; i++)
  {
     for (j=i+1; j < N; j++)
     {
       if (strcmp(Student[i].Name, Student[j].Name) > 0)
       {
         // Swap ith student with jth student:
         tmp = Student[j];
         Student[j] = Student[i];
         Student[i] = tmp;
       }
     }
  }

  cout << "\nLIST SORTED BY NAME:\n";
  for (i=0; i < N; i++)
     cout << setw(11) << Student[i].Name << " " << Student[i].Age << "\n";

  cout << "\n";
  system("PAUSE");
  return 0;
}
```