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

//This program gets up to 10 scores from a students, calculates the average and final grade
//Security: NO INPUT SANITIZATION, TYPE CHECKING AT ALL

void GetStrInput(char[], int);
void GetIdInput(char[], int);
void GetComments(char[], int);

int GetNDigitInput(int);

int main()
{
 //------DO NOT CHANGE THESE------
 int scores[10];
 char name[100] = "";
 char id[12] = "";
 int number_of_score = 0;
 double average = 0;
 int sum = 0;
 char grade = 'X';
 char comments[5] = "NONE";

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

 cout << "\n---------------------------------------\n";
 cout << "Welcome to the scoring system!" <<endl;

 cout << "Enter student's name: ";  //BUFFER OVERFLOW FIXED
 GetStrInput(name, sizeof(name));

 cout << "\nPlease enter student ID: "; //BUFFER OVERFLOW FIXED
 GetIdInput(id, 10 * sizeof(char));

 cout << "Enter up to 10 student's scores (0-100) (if done, enter -1 to stop): ";
 cout << endl;
 for (int i = 0; i < 10; i++)
 {
  cout << "Score " << i+1 << ": ";
  int temp;
  temp = GetNDigitInput(3); //fixed possibility for character entry screwing up system

  if (temp != -1)
  {
   scores[i] = temp;
   number_of_score++;
   sum = sum + temp;
  }

  else
  {
   break;
  }
 }

 if(number_of_score > 0)
 {
  average = sum / number_of_score;
 }

 else
 {
  average = 0;
 }
```

```
 cout << "Student's name is: " << name << endl;
 cout << "Student has " << number_of_score << " scores, sum is " << sum <<", and the average
score is " << average << endl;

 // figure out the final grade, A, B, C, D, or F
 if( average >= 90)
  grade = 'A';
 if(average < 90 and average >= 80)
  grade = 'B';
 if (average < 80 and average >= 70)
  grade = 'C';
 if (average < 70 and average >= 60)
  grade = 'D';
 if (average < 60 and average >= 0)
  grade = 'F';

 cout << "Based on the average score, final grade is: " << grade << endl;

 cout << "looks good? (Yes or No): ";

 GetComments(comments, sizeof(comments)); //BUFFER OVERFLOW

 cout << "Comments - Looks good? - " << comments << endl << endl;
 cout << "Student name: " << name << endl;
 cout << "Student ID: " << id <<endl;
 cout << "Final grade is: " << grade << endl <<endl;

 cout << "Program exits successfully..." <<endl;
}

int GetNDigitInput(int n)
{
 bool isValidInput = false;

 char tmpStr[n + 1];
 int  tmpInt = 0;

 while(!isValidInput)
 {
  isValidInput = true;

  //get input
  GetStrInput(tmpStr, sizeof(tmpStr));

  //validate size
  if(strnlen(tmpStr, sizeof(tmpStr)) > n)
  {
   isValidInput = false;
  }

  //handle early exit
  if(strncmp(tmpStr, "-1", sizeof(tmpStr)) == 0)
  {
   return -1;
  }

  //validate string and turn string into int
  for(int i = 0; tmpStr[i] != '\0'; i++)
  {
   if(tmpStr[i] < '0' || tmpStr[i] > '9')
   {
    isValidInput = false;
   }

   tmpInt = tmpInt * 10 + (tmpStr[i] - '0');
```

```
  }

  if(!isValidInput)
  {
   printf("Input must be up to 3 digits, and numeric only. Try again: ");
   tmpInt = 0;
  }
 }

 return tmpInt;
}

void GetIdInput(char str[], int size)
{
 bool gotValidInput = false;

 while(!gotValidInput)
 {
  gotValidInput = true;

  GetStrInput(str, size);

  //ensure format of xxx-xxxxx
  for(int i = 0; i < size - 1; i++)
  {
   //ignore the '-'
   if(i == 3)
   {
    i++;
   }

   //if character is non-numeric, invalidate the input
   if(str[i] < '0' || str[i] > '9')
   {
    gotValidInput = false;
    break;
   }
  }

  //check for '-' on char 4
  if(str[3] != '-')
  {
   gotValidInput = false;
  }

  if(!gotValidInput)
  {
   printf("Invalid Input, try again: ");
  }
 }
}

void GetStrInput(char str[], int size)
{
 bool gotValidInput = false;

 while(!gotValidInput)
 {
  // get input
  fgets(str, size, stdin);

  //strip newline char if it is present
  for(int i = 0; i < size; i++)
  {
   if(str[i] == '\n')
   {
    str[i] = '\0';
```

```
    //break;
   }
  }

  //if string input is valid, set flag and exit loop
  int inputSize = strnlen(str, size);

  if(inputSize > 0)
  {
   gotValidInput = true;

   //clear input buffer if max size is reached
   if(inputSize >= size - 1)
   {
    while (getchar() != '\n');
   }
  }

  //otherwise, continue to loop
  else
  {
   printf("Invalid input, try again: ");
  }
 }
}

void GetComments(char str[], int size)
{
 bool gotValidInput = false;
 while(!gotValidInput)
 {
  //get up to size characters of input
  for(int i = 0; i < size; i++)
  {
   str[i] = (char)getchar();

   //end early if enter pressed early and strip new line
   if(str[i] == '\n')
   {
    str[i] = '\0';
    break;
   }
  }

  //attach null terminator
  str[size - 1] = '\0';

  int inputSize = strnlen(str, size);

  //check for valiid input
  if(inputSize > 0)
  {
   gotValidInput = true;

   //clear input buffer if max size is reached
   if(inputSize >= size - 1)
   {
    while (getchar() != '\n');
   }
  }

  //invalid input
  else
  {
   gotValidInput = false;
   printf("Invalid input, try again: ");
  }
```

```
  }
}
```