**Eastern Mediterranean University**



**Department of Computer Engineering**

**Famagusta, North Cyprus**

**Project Report CMSE 326**

**Development of the Software Boundary Value Testing Tool (SBVTT)**

**Project 2**

**Students: Group 8**

1. Talal Hatem Mahdy (147139)
2. Ahmet Sezen (142323)
3. Mert Kasar (120938)

**Instructor:** Assoc. Prof. Dr. Alexander Chefranov  
**Lab Coordinator:** Kamran Kamaei

**May 2016**

**Outline:**

The aim of this project is to develop a software testing tool which allows us to test any given program using any of the following testing techniques:  
1. Normal Boundary Value Testing.  
2. Robust Boundary Value Testing.  
3. Worst Case Normal Boundary Value Testing.  
4. Worst Case Robust Boundary Value Testing.  
 **Problem Definition:**

The SBVTT is to be tested on the following programs:  
1. Next Date Program.  
2. Day of Week Program.  
3. Triangle Program.  
A report containing all the actions taken and results gained after the user finished using the tool should be generated.

**Project Design:**

The SBVTT is designed as a menu driven program which can access any other program and test it. The design flow of the tool with an example goes like this:  
1. Welcome to Software Boundary Value Testing Tool (SBVTT)  
 What do you want to do?  
 1. Generate Test Cases (inputs.txt)   
 2. Generate outputs (outputs.txt)  
 3. Test the programs by verifying the outputs   
 4. Exit and save report (report.txt)   
 (user selects 1)

2. How many variables does your program contain? (user enters 3)

3. Choose a testing method from the following: (user selects 1)  
 1. Normal Boundary Value Testing   
 2. Robust Boundary Value Testing   
 3. Worst Case Normal Boundary Value Testing   
 4. Worst Case Robust Boundary Value Testing

4. Enter minimum and maximum input values for your 3 variables  
 Variable 1: (user enters 1 31)  
 Variable 2: (user enters 1 12)  
 Variable 3: (user enters 1812 2012)

5. Generating test cases for your chosen test method…  
 They will be saved in the project folder as “inputs.txt”

6. (“What do you want to do?” menu reappears and the user presses 2 and  
 selects to generate outputs and save them to output.txt)

7. (“What do you want to do?” menu reappears and the user presses 3 and  
 selects to Test one of the Programs)  
 Choose a program to be tested from the following:  
 1. Next date program 2. Day of week program 3. Triangle Program  
 (user selects option 1 to test the nextdate program)

8. (User will be asked if each displayed output is correct with respect to  
 its input)  
 Now we have a ready to use input and output files. Let us confirm if the  
 outputs are correct. Please click 1 if the displayed output is wrong to  
 calculate the mistakes out of all input test cases. Press any other key  
 if the displayed output is correct.  
  
 Input 1: 1 6 1912  
 Output 1: 2 6 1912  
 (user presses any key other than 1)  
  
 16 6 2011  
 17 6 2011   
 (user presses any key other than 1)  
 …  
 Error = (number of errors discovered by user/number of test cases) \* 100  
 (The error in the user’s program is displayed)  
9. (“What do you want to do?” menu reappears and the user presses 4 and  
 selects to exit the tool and save the automatically generated report on   
 report.txt. This report contains all the actions taken and results gained  
 after the user finished using the tool)

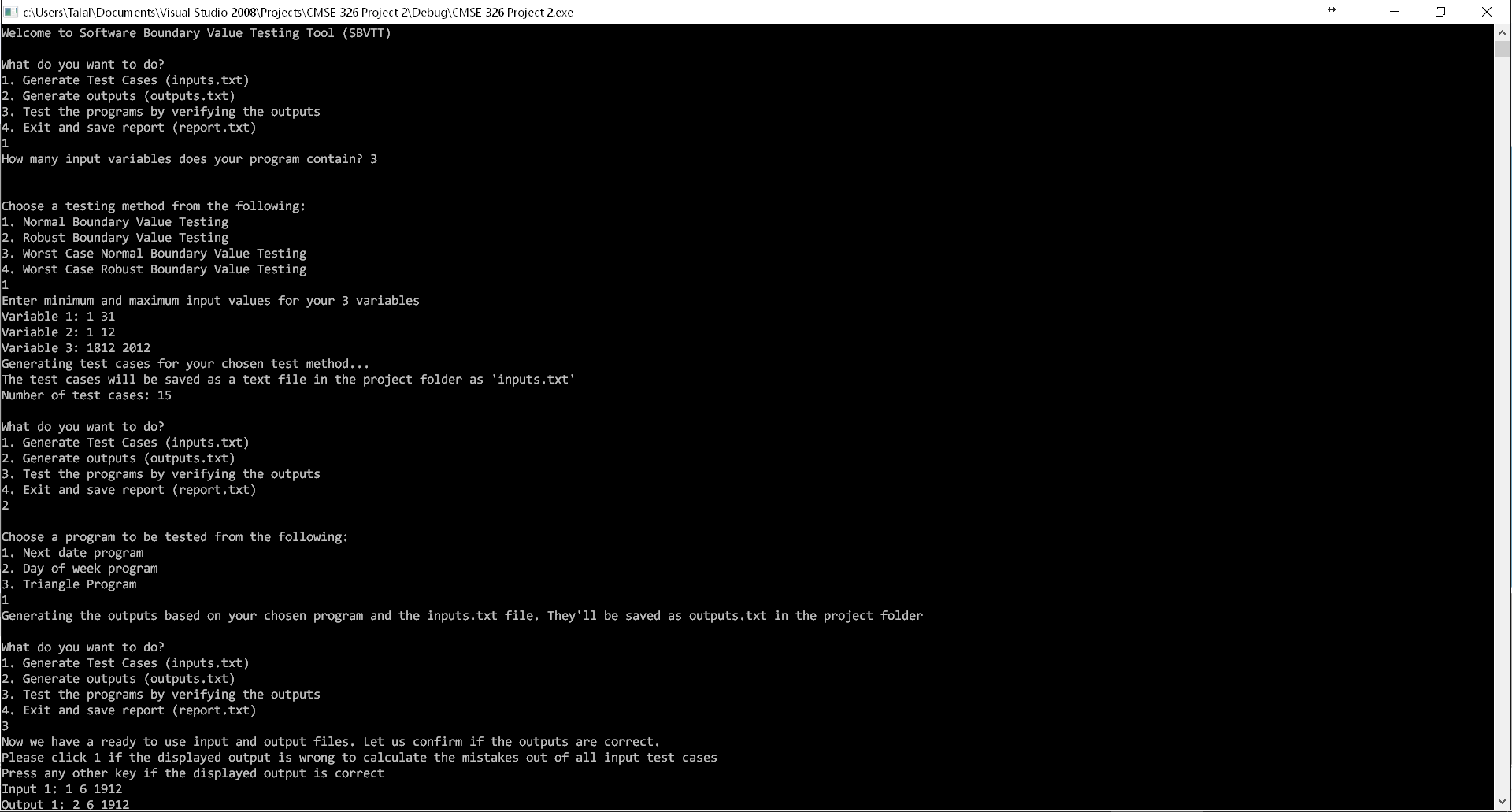
**Work Breakdown:**

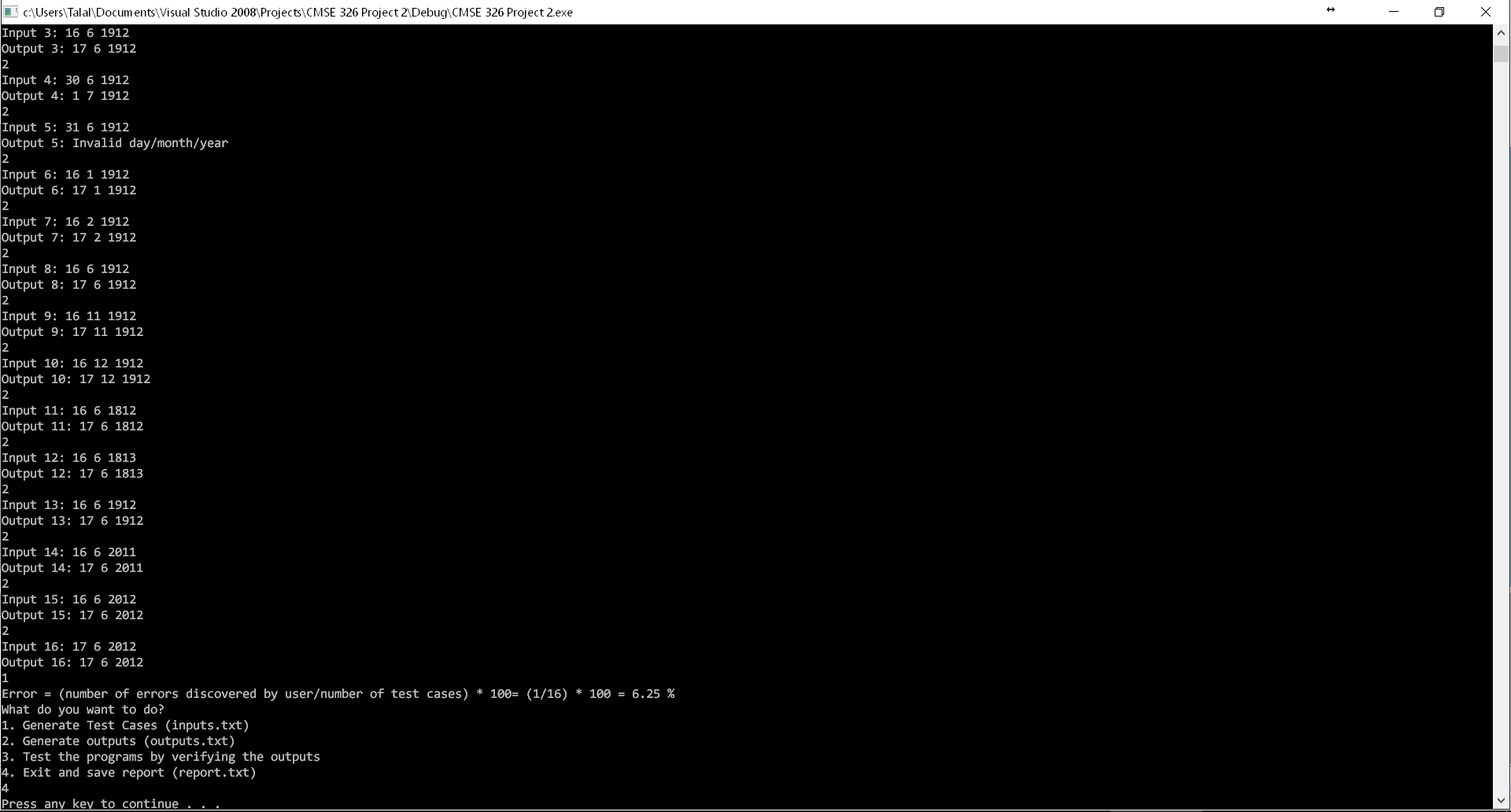
All team members cooperated together to complete all units of the program.   
1st Week: Meet each other; Begin Project; Input stage  
2nd Week: Output stage  
3rd Week: Testing stage  
4th Week: Discuss project; Implement all units and test the SBVTT; Prepare the  
 final report, presentation and CD.

**Project Implementation:**The SBVTT was implemented using C++ as a programming language.  
Since this project was developed using Microsoft Visual C++, the runtime libraries for Visual C++ should be installed on the user’s computer to run applications developed using Visual Studio.   
Also, if a user wishes to test his program, the .exe file of his program should be located in the same location as the .exe file of the SBVTT. The inputs.txt, outputs.txt and report.txt files will also be generated in the same location.  
This tool relies heavily on the user’s inputs to provide the required information to accomplish his needs of testing his program. Therefore, a menu-driven design was implemented throughout the tool (See Appendix A-D).   
The tool mainly composes of 3 units. Input stage, output stage, testing stage and report generation stage.   
This tool should also store all the input and output data and an automatically generated report on text files. Therefore, the usage of data files concept was largely implemented in the tool.   
The concepts of arrays, pointers and dynamic memory allocation were also implemented in the generation of input and output cases.   
The most important concept in this tool is the implementation of the technique to make it possible to use this tool to test any given program. This can be done using many different methods but we implemented it using the command system(const char \*); combined with modifying the main function in the tested program so that it could take two arguments, namely int argc and   
char \*argv[]. We could then convert the argv argument into the inputs of the program to be tested using the method of parsing to int.

**Project Testing:**

The tool underwent rigorous and robust testing to make sure every unit in the tool works as how it was meant to work. The tool was tested by entering all possible inputs that the user could give to the tool. Below are some screenshots demonstrating the debugging of the tool.

****

****

**Conclusion:**

It is obvious that the SBVTT can be very useful since it can save us a lot of time when we want to test a program. It can generate all input and output cases of a program in a few seconds. It can also test our program using these test cases and generate a useful report showing the user what actions he took while using the tool and the results gained from it. We learned a lot of things during the development of this project. We learnt how to handle data and work with large text files. We also got more experience by learning how to generate inputs for 4 various types of tests using the concepts of loops, arrays, pointers, dynamic memory allocation, etc. We also learnt a whole new concept which is executing a program using another program.

*References*

Arora, H. (2016). *C argc and argv Examples to Parse Command Line Arguments*. *Thegeekstuff.com*. Retrieved 22 May 2016, from http://www.thegeekstuff.com/2013/01/c-argc-argv/

*C - itoa() function*. *Fresh2refresh.com*. Retrieved 22 May 2016, from http://fresh2refresh.com/c-programming/c-type-casting/c-itoa-function/

*C library function - memset()*. *www.tutorialspoint.com*. Retrieved 22 May 2016, from http://www.tutorialspoint.com/c\_standard\_library/c\_function\_memset.htm

*C Program to Print Current Date and Time*. (2016). *Techcrashcourse.com*. Retrieved 22 May 2016, from http://www.techcrashcourse.com/2014/10/c-program-to-print-current-date-time.html

*Run a program from within a program*. (2003). *Faq.cprogramming.com*. Retrieved 22 May 2016, from http://faq.cprogramming.com/cgi-bin/smartfaq.cgi?answer=1044654269&id=1043284392

*system function*. (2004). *Pubs.opengroup.org*. Retrieved 22 May 2016, from http://pubs.opengroup.org/onlinepubs/009695399/functions/system.html

*The GNU C Programming Tutorial*. *Crasseux.com*. Retrieved 22 May 2016, from http://crasseux.com/books/ctutorial/argc-and-argv.html

**APPENDIX A**

**SBVTT Source Code:**

#include<iostream>

#include<stdio.h>

#include<time.h>

#define N 10000

using namespace std;

int finished=0;

int t;

void updatejarr(int jarr[],int size)

{

if(size==0)

{

finished=1;

return;

}

if(t==3)

{

if(jarr[size-1]==4)

{

jarr[size-1]=0;

updatejarr(jarr,size-1);

}

else

jarr[size-1]++;

}

else //t==4

{

if(jarr[size-1]==6)

{

jarr[size-1]=0;

updatejarr(jarr,size-1);

}

else

jarr[size-1]++;

}

}

/////////////////////////// MAIN ///////////////////////////////// MAIN ////////////////////////////////////// MAIN /////////////////////////////////

int main()

{

int choice;

int n, i, j, k, p, counter=0, error=0, errorchoice=0;

int a,b,c; //inputs/outputs

int \*jarr;

int \*\*mat;

float totalerror;

char string[20];

char str[N], buffer1[N],buffer2[N],buffer3[N];

FILE \*inputs,\*output,\*report;

time\_t current\_time; //to print the date and time in the report

char\* c\_time\_string;

current\_time = time(NULL);

c\_time\_string = ctime(&current\_time);

report=fopen("report.txt","w");

fprintf(report,"Report generated on %s\n",c\_time\_string);

cout<<"Welcome to Software Boundary Value Testing Tool (SBVTT)\n";

do

{

counter=0;

cout<<"\nWhat do you want to do?\n1. Generate Test Cases (inputs.txt)\n2. Generate outputs (outputs.txt)\n";

cout<<"3. Test the programs by verifying the outputs\n4. Exit and save report (report.txt)\n";

cin>>choice;

if(choice==1)

{

inputs = fopen("inputs.txt","w");

fprintf(report,"The user chose to generate input variables and store them in inputs.txt\n");

cout<<"How many input variables does your program contain? ";

cin>>n;

cout<<endl;

fprintf(report,"The user's program contained %d variables\n",n);

do

{

cout<<endl<<"Choose a testing method from the following: \n";

cout<<"1. Normal Boundary Value Testing\n2. Robust Boundary Value Testing\n3. Worst Case Normal Boundary Value Testing\n4. Worst Case Robust Boundary Value Testing\n";

cin>>t;

if(t!=1 && t!=2 && t!=3 && t!=4)

cout<<"Please choose a testing method from 1-4\n";

}while(t!=1 && t!=2 && t!=3 && t!=4); //t<1 || t>4

////////////////////// INPUT VARIABLES //////////////////////////// INPUT VARIABLES //////////////////////////// INPUT VARIABLES /////////////////////////

cout<<"Enter minimum and maximum input values for your "<<n<<" variables\n";

if(t==1)

{

fprintf(report,"The user chose to test his program using the Normal Boundary Value Testing method\n");

mat=new int\*[n];

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

mat[i]=new int[5];

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

{

cout<<"Variable "<<i+1<<": ";

cin>>mat[i][0]>>mat[i][4];

mat[i][1]=mat[i][0]+1;

mat[i][3]=mat[i][4]-1;

mat[i][2]=(mat[i][0]+mat[i][4])/2;

}

}

if(t==2)

{

fprintf(report,"The user chose to test his program using the Robust Boundary Value Testing method\n");

mat=new int\*[n];

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

mat[i]=new int[7];

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

{

cout<<"Variable "<<i+1<<": ";

cin>>mat[i][1]>>mat[i][5];

mat[i][2]=mat[i][1]+1;

mat[i][4]=mat[i][5]-1;

mat[i][3]=(mat[i][1]+mat[i][5])/2;

mat[i][0]=mat[i][1]-1;

mat[i][6]=mat[i][5]+1;

}

}

if(t==3)

{

fprintf(report,"The user chose to test his program using the Worst Case Normal Boundary Value Testing method\n");

jarr=new int[n];

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

jarr[i]=0;

mat=new int\*[n];

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

mat[i]=new int[5];

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

{

cout<<"Variable "<<i+1<<": ";

cin>>mat[i][0]>>mat[i][4];

mat[i][1]=mat[i][0]+1;

mat[i][3]=mat[i][4]-1;

mat[i][2]=(mat[i][0]+mat[i][4])/2;

}

}

if(t==4)

{

fprintf(report,"The user chose to test his program using the Worst Case Robust Boundary Value Testing method\n");

jarr=new int[n];

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

jarr[i]=0;

mat=new int\*[n];

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

mat[i]=new int[7];

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

{

cout<<"Variable "<<i+1<<": ";

cin>>mat[i][1]>>mat[i][5];

mat[i][2]=mat[i][1]+1;

mat[i][4]=mat[i][5]-1;

mat[i][3]=(mat[i][1]+mat[i][5])/2;

mat[i][0]=mat[i][1]-1;

mat[i][6]=mat[i][5]+1;

}

}

//////////////// GENERATE TEST CASES ////////////////////////// GENERATE TEST CASES //////////////////// GENERATE TEST CASES ////////////////////////

cout<<"Generating test cases for your chosen test method...\nThe test cases will be saved as a text file in the project folder as 'inputs.txt'\n";

if(t==1)

{

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

{

for(j=0;j<5;j++)

{

for(k=0;k<n;k++)

{

if(k==i)

fprintf(inputs,"%d", mat[k][j]);

else

fprintf(inputs,"%d", mat[k][2]);

fprintf(inputs," " );

}

fprintf(inputs,"\n" );

counter++;

}

}

}

if(t==2)

{

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

{

for(j=0;j<7;j++)

{

for(k=0;k<n;k++)

{

if(k==i)

fprintf(inputs,"%d", mat[k][j]);

else

fprintf(inputs,"%d", mat[k][3]);

fprintf(inputs," " );

}

fprintf(inputs,"\n" );

counter++;

}

}

}

if(t==3)

{

while(!finished)

{

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

{

j=jarr[i];

fprintf(inputs,"%d ", mat[i][j]);

}

updatejarr(jarr,n);

counter++;

if(!finished)fprintf(inputs,"\n" );

}

}

if(t==4)

{

while(!finished)

{

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

{

j=jarr[i];

fprintf(inputs,"%d ", mat[i][j]);

}

updatejarr(jarr,n);

counter++;

if(!finished)fprintf(inputs,"\n" );

}

}

cout<<"Number of test cases: "<<counter<<endl;

fprintf(report,"%d Test cases were generated by the user and stored in inputs.txt\n",counter);

fclose(inputs);

}

/////////////////// GENERATING OUTPUTS ////////////////////// GENERATING OUTPUTS ////////////////////// GENERATING OUTPUTS ///////////////////////////

else if(choice==2)

{

output=fopen("outputs.txt","w"); //clearing the output file before generating

fclose(output);

inputs=fopen("inputs.txt","r");

do

{

cout<<endl<<"Choose a program to be tested from the following: \n";

cout<<"1. Next date program\n2. Day of week program\n3. Triangle Program\n";

cin>>p;

if(p!=1 && p!=2 && p!=3)

cout<<"Please choose a program from 1-3\n";

}while(p!=1 && p!=2 && p!=3);

cout<<"Generating the outputs based on your chosen program and the inputs.txt file. They'll be saved as outputs.txt in the project folder\n";

do

{

fscanf(inputs,"%d %d %d", &a,&b,&c);

itoa(a,buffer1,10); //converting int to const char\*

itoa(b,buffer2,10);

itoa(c,buffer3,10);

if(p==1)

strcpy(str,"nextdate.exe 3 ");

if(p==2)

strcpy(str,"dayofweek.exe 3 ");

if(p==3)

strcpy(str,"triangle.exe 3 ");

strcat(str,buffer1);

strcat(str," ");

strcat(str,buffer2);

strcat(str," ");

strcat(str,buffer3);

system(str); //executing the program //using argc and argv to send inputs to the programs

memset(&buffer1[0], 0, sizeof(buffer1)); //setting all elements in the string to 0

memset(&buffer2[0], 0, sizeof(buffer2));

memset(&buffer3[0], 0, sizeof(buffer3));

}while(feof(inputs)==0);

fprintf(report,"The outputs of input.txt file were generated by the user and stored in outputs.txt\n");

fclose(inputs);

}

//////////////// OUTPUT VALIDATION ////////////////// OUTPUT VALIDATION //////////////////////// OUTPUT VALIDATION //////////////////////////////////

else if(choice==3)

{

inputs=fopen("inputs.txt","r");

output=fopen("outputs.txt","r");

cout<<"Now we have a ready to use input and output files. Let us confirm if the outputs are correct.\n";

cout<<"Please click 1 if the displayed output is wrong to calculate the mistakes out of all input test cases\n";

cout<<"Press any other key if the displayed output is correct\n";

counter=0;

if(p==1)

{

do

{

fscanf(inputs,"%d %d %d", &a,&b,&c);

cout<<"Input "<<counter+1<<": "<<a<<" "<<b<<" "<<c<<endl;

fscanf(output,"%d %d %d", &a,&b,&c);

if(a==0 && b==0 && c==0)

cout<<"Output "<<counter+1<<": Invalid day/month/year"<<endl;

else

cout<<"Output "<<counter+1<<": "<<a<<" "<<b<<" "<<c<<endl;

cin>>errorchoice;

if(errorchoice==1)

error++;

counter++;

}while(feof(inputs)==0);

fprintf(report,"The user chose to test the nextdate program using the generated outputs\n");

}

if(p==2)

{

cout<<"1. Sunday 2. Monday 3. Tuesday 4. Wednesday 5. Thursday 6. Friday 7. Saturday\n";

do

{

fscanf(inputs,"%d %d %d", &a,&b,&c);

cout<<"Input "<<counter+1<<": "<<a<<" "<<b<<" "<<c<<endl;

fscanf(output,"%d", &a);

cout<<"Output "<<counter+1<<": "<<a<<endl;

cin>>errorchoice;

if(errorchoice==1)

error++;

counter++;

}while(feof(inputs)==0);

fprintf(report,"The user chose to test the dayofweek program using the generated outputs\n");

}

if(p==3)

{

do

{

fscanf(inputs,"%d %d %d", &a,&b,&c);

cout<<"Input "<<counter+1<<": "<<a<<" "<<b<<" "<<c<<endl;

fscanf(output,"%s", string);

cout<<"Output "<<counter+1<<": "<<string<<endl;

cin>>errorchoice;

if(errorchoice==1)

error++;

counter++;

}while(feof(inputs)==0);

fprintf(report,"The user chose to test the triangle program using the generated outputs\n");

}

totalerror=(float)error/counter \* 100;

cout<<"Error = (number of errors discovered by user/number of test cases) \* 100= ("<<error<<"/"<<counter<<") \* 100 = "<<totalerror<<" %";

fprintf(report,"The error percentage in the user's tested program is %f %\n",totalerror);

fclose(inputs);

fclose(output);

}

else if(choice==4)

break;

}while(1);

fclose(report);

system("pause");

return 0;

}

**APPENDIX B**

**Next date Source Code:**

#include<iostream>

#include<stdio.h>

using namespace std;

int parseint(char\* s);

int getdigit(char c);

int parseint(char\* s)

{

int i,n=0;

for(i=0;i<strlen(s);i++)

n=n\*10+getdigit(s[i]);

return n;

}

int getdigit(char c)

{

switch(c)

{

case '0':return 0;

case '1':return 1;

case '2':return 2;

case '3':return 3;

case '4':return 4;

case '5':return 5;

case '6':return 6;

case '7':return 7;

case '8':return 8;

case '9':return 9;

};

}

int main(int argc, char\*\* argv)

{

//puts(\*argv);

int day, month, year;

day=parseint(argv[2]);

month=parseint(argv[3]);

year=parseint(argv[4]);

int tomorrowday,tomorrowmonth,tomorrowyear;

bool c=1,c1,c2,c3;

bool leap;

FILE \*outputs = fopen("outputs.txt","a");

c1=(1<=day) && (day<=31);

c2=(1<=month) && (month<=12);

c3=(1812<=year) && (year<=2012);

if(c1&&c2&&c3)

c=0;

else

c=1;

if((year%100==0 && year%400==0)||(year%100!=0 && year%4==0))

leap=1;

else

leap=0;

tomorrowmonth=month;

tomorrowyear=year;

if(month==1||month==3||month==5||month==7||month==8||month==10)

{

if (day<31)

tomorrowday=day+1;

else

{

tomorrowday=1;

tomorrowmonth=month+1;

}

}

else if(month==4||month==6||month==9||month==11)

{

if (day<30)

tomorrowday=day+1;

else if (day==30)

{

tomorrowday=1;

tomorrowmonth=month+1;

}

else

c=0;

}

else if(month==12)

{

if(day<31)

tomorrowday=day+1;

else

{

tomorrowday=1;

tomorrowmonth=1;

tomorrowyear=year+1;

}

}

else //february

{

if((day<28)||(day==28 && leap==1))

tomorrowday=day+1;

else if((day==28 && leap==0)||(day==29 && leap==1))

{

tomorrowday=1;

tomorrowmonth=3;

}

else

c=0;

}

if(c==0)

{

tomorrowday=0;tomorrowmonth=0;tomorrowyear=0; //invalid input

fprintf(outputs,"%d %d %d\n",tomorrowday,tomorrowmonth,tomorrowyear);

}

if(c==1)

fprintf(outputs,"%d %d %d\n",tomorrowday,tomorrowmonth,tomorrowyear);

fclose(outputs);

}

**APPENDIX C**

**Day of week Source Code:**

#include<iostream>

#include<stdio.h>

using namespace std;

int parseint(char\* s);

int getdigit(char c);

int parseint(char\* s)

{

int i,n=0;

for(i=0;i<strlen(s);i++)

n=n\*10+getdigit(s[i]);

return n;

}

int getdigit(char c)

{

switch(c)

{

case '0':return 0;

case '1':return 1;

case '2':return 2;

case '3':return 3;

case '4':return 4;

case '5':return 5;

case '6':return 6;

case '7':return 7;

case '8':return 8;

case '9':return 9;

};

}

int main(int argc, char\*\* argv)

{

//puts(\*argv);

int day, month, year;

int dayofweek, k, j;

FILE \*outputs = fopen("outputs.txt","a");

day=parseint(argv[2]);

month=parseint(argv[3]);

year=parseint(argv[4]);

//day=1;

//month=1;

//year=2016;

if(month<3)

{

month += 12;

year -= 1;

}

k = year% 100;

j = year/100;

dayofweek = ((day + (((month + 1) \* 26)/10) + k + (k/4) + (j/4)) + (5 \* j))%7;

fprintf(outputs,"%d\n",dayofweek);

fclose(outputs);

}

**APPENDIX D**

**Triangle Source Code:**

#include<iostream>

#include<stdio.h>

using namespace std;

int parseint(char\* s);

int getdigit(char c);

int parseint(char\* s)

{

int i,n=0;

for(i=0;i<strlen(s);i++)

n=n\*10+getdigit(s[i]);

return n;

}

int getdigit(char c)

{

switch(c)

{

case '0':return 0;

case '1':return 1;

case '2':return 2;

case '3':return 3;

case '4':return 4;

case '5':return 5;

case '6':return 6;

case '7':return 7;

case '8':return 8;

case '9':return 9;

};

}

int main(int argc, char\*\* argv)

{

//puts(\*argv);

int a,b,c,match=0;

int result;

bool cond=1,c1,c2,c3,c4,c5,c6;

FILE \*outputs = fopen("outputs.txt","a");

a=parseint(argv[2]);

b=parseint(argv[3]);

c=parseint(argv[4]);

c1=(1<=a) && (a<=200);

c2=(1<=b) && (b<=200);

c3=(1<=c) && (c<=200);

c4=a<(b+c);

c5=b<(a+c);

c6=c<(a+b);

if(!(c1&&c2&&c3))

{

fprintf(outputs,"Out of range\n");

goto end;

}

if(!(c4||c5||c6))

{

fprintf(outputs,"NotATriangle\n");

goto end;

}

if (a == b)

match = match + 1;

if (a == c)

match = match + 2;

if (b == c)

match = match + 3;

if (match == 0)

{

if ((a + b)<= c)

fprintf(outputs,"NotATriangle\n");

else if((b+c)<=a)

fprintf(outputs,"NotATriangle\n");

else if((a+c)<=b)

fprintf(outputs,"NotATriangle\n");

else

fprintf(outputs,"Scalene\n");

}

else if(match==1)

{

if((a+c)<=b)

fprintf(outputs,"NotATriangle\n");

else

fprintf(outputs,"Isosceles\n");

}

else if(match==2)

{

if((a+c)<=b)

fprintf(outputs,"NotATriangle\n");

else

fprintf(outputs,"Isosceles\n");

}

else if(match==3)

{

if((b+c)<=a)

fprintf(outputs,"NotATriangle\n");

else

fprintf(outputs,"Isosceles\n");

}

else

fprintf(outputs,"Equilateral\n");

end:

fclose(outputs);

}