**MINI PROJECT REPORT**

**PRANJAL SHRIVASTAVA**

**NOVEMBER 21, 2022**

ABSTRACT

This report contains the information about my mini project in C language, named “ONLINE COMPUTER QUIZ”

**Contents**

1. Introduction
2. Function Description
3. Code in C
4. Profiling
5. Debugging

**1.Introduction**

I made a computer quiz program in C language. It contains 10 basic computer based question. In which the person can check his/her knowledge about the programming and computer basics.

**2.Function information**

Void menu()is the function which gives general information about attempt the questions.

Void question() is the function used to print each and every question and provide information to the user that whether the answer given by user is correct or not.

Void calculator() is the function which calculates the marks obtained by user for every question. 10 marks will be allotted for every correct question and there is no negative marking. And this function add the total marks.

Char result() is the function which shows the result that whether the answer given by user is correct or not.

#include<stdio.h>

#include<stdlib.h>

char name[30];

int userScore=0,countQ=0;

void calculateScore()

{

if (userScore >= 80&&userScore <= 100)

{

printf("\nCongrats! %s You are win the quiz.\n",name);

printf("you got %d numbers.\n",userScore);

printf("your %d Question in correct.\n",countQ);

}

else if(userScore >= 60&&userScore < 80)

{

printf("Congrats! %s You are win the quiz.\n",name);

printf("you got %d numbers.\n",userScore);

printf("your %d Question in correct.\n",countQ);

}

else if(userScore >= 40&&userScore < 60)

{

printf("Congrats! %s You are win the quiz.\n",name);

printf("you got %d numbers.\n",userScore);

printf("your %d Question in correct.\n",countQ);

}

else if(userScore >= 10&&userScore < 40)

{

printf("Sorry! %s You are loose the quiz.\n",name);

printf("you got %d numbers.\n",userScore);

printf("your %d Question in correct.\n",countQ);

printf("\*\*Better luck lext time\*\*\*\*\n");

}

exit(0);

}

char result(char choose,char correct)

{

char next;

if (choose==correct)

{

countQ++;

userScore=userScore+10;

printf("Answer is correct!\n");

printf("Press (Y) to continue Quiz ");

printf("If you want to end this Quiz then press (N)\n");

fflush(stdin);

scanf("%c",&next);

return(next);

}else

{

printf("Sorry Answer is wrong!\n");

printf("Press (Y) to continue Quiz ");

printf("If you want to end this Quiz then press (N)\n");

fflush(stdin);

scanf("%c",&next);

return(next);

}

}

void question()

{

char choose,correct;

printf("1.Q-which data type store characters?\n");

printf("\t(A).Int\t(B).float\n");

printf("\t(C).char\t(D).byte\n");

printf("Choose which option is correct:");

fflush(stdin);

scanf("%c",&choose);

correct='c';

correct=result(choose,correct);

if (correct=='n'||correct=='N')

{

return;

}

printf("2.Q-How many bytes consume Int data type in 64 bit OS?\n");

printf("\t(A).4\t(B).6\n");

printf("\t(C).2\t(D).8\n");

printf("Choose which option is correct:");

fflush(stdin);

scanf("%c",&choose);

correct='a';

correct=result(choose,correct);

if (correct=='n'||correct=='N')

{

return;

}

printf("3.Q-How many bytes consume Float data type in 64 bit OS?\n");

printf("\t(A).6\t(B).4\n");

printf("\t(C).2\t(D).8\n");

printf("Choose which option is correct:");

fflush(stdin);

scanf("%c",&choose);

correct='b';

correct=result(choose,correct);

if (correct=='n'||correct=='N')

{

return;

}

printf("4.Q-How many bytes consume Double data type in 64 bit OS?\n");

printf("\t(A).4\t(B).6\n");

printf("\t(C).2\t(D).8\n");

printf("Choose which option is correct:");

fflush(stdin);

scanf("%c",&choose);

correct='d';

correct=result(choose,correct);

if (correct=='n'||correct=='N')

{

return;

}

printf("5.Q-How many bytes consume char data type in 64 bit OS?\n");

printf("\t(A).4\t(B).6\n");

printf("\t(C).1\t(D).8\n");

printf("Choose which option is correct:");

fflush(stdin);

scanf("%c",&choose);

correct='c';

correct=result(choose,correct);

if (correct=='n'||correct=='N')

{

return;

}

printf("6.Q-Which type of values is store in Int data type?\n");

printf("\t(A).integer\t(B).floating point\n");

printf("\t(C).character\t(D).string\n");

printf("Choose which option is correct:");

fflush(stdin);

scanf("%c",&choose);

correct='a';

correct=result(choose,correct);

if (correct=='n'||correct=='N')

{

return;

}

printf("7.Q-Which type of values is store in float data type?\n");

printf("\t(A).integer\t(B).floating point\n");

printf("\t(C).character\t(D).string\n");

printf("Choose which option is correct:");

fflush(stdin);

scanf("%c",&choose);

correct='b';

correct=result(choose,correct);

if (correct=='n'||correct=='N')

{

return;

}

printf("8.Q-Which type of values is store in double data type?\n");

printf("\t(A).integer\t(B).floating point\n");

printf("\t(C).character\t(D).string\n");

printf("Choose which option is correct:");

fflush(stdin);

scanf("%c",&choose);

correct='b';

correct=result(choose,correct);

if (correct=='n'||correct=='N')

{

return;

}

printf("9.Q-What is string?\n");

printf("\t(A).Integer values\t(B).floating values\n");

printf("\t(C).Character array\t(D).Array\n");

printf("Choose which option is correct:");

fflush(stdin);

scanf("%c",&choose);

correct='c';

correct=result(choose,correct);

if (correct=='n'||correct=='N')

{

return;

}

printf("10.Q-What is structure?\n");

printf("\t(A).Integer\t(B).float\n");

printf("\t(C).Character\t(D).User defined data type\n");

printf("Choose which option is correct:");

fflush(stdin);

scanf("%c",&choose);

correct='d';

correct=result(choose,correct);

if (correct=='n'||correct=='N')

{

return;

}

calculateScore();

}

void menu()

{

char ch;

printf("\nHello! Mister %s\n",name);

printf("Here are some rules of this Game.\n");

printf("1. You can choose any option.\n");

printf("2. You need to answer 10 question\n");

printf("3. Every question is 10 number\n");

printf("4.Total number is 100\n");

printf("5. We decide you win the quiz or not\n");

printf("Press (Y) for start quiz otherwise press (N)\n");

fflush(stdin);

scanf("%c",&ch);

if(ch=='y'||ch=='Y')

{

question();

calculateScore();

}else

{

exit(0);

}

}

void main()

{

printf("\n\*\*\*\*Welcome to Quiz Game\*\*\n");

printf("Enter your name:");

gets(name);

menu();

}

**OUTPUT OF PROGRAM---**

PS C:\Users\shriv\Downloads\pranjal> .\miniproject

\*\*\*\*Welcome to Quiz Game\*\*

Enter your name:pranjal

Hello! Mister pranjal

Here are some rules of this Game.

1. You can choose any option.

2. You need to answer 10 question

3. Every question is 10 number

4.Total number is 100

5. We decide you win the quiz or not

Press (Y) for start quiz otherwise press (N)

y

1.Q-which data type store characters?

(A).Int (B).float

(C).char (D).byte

Choose which option is correct:a

Sorry Answer is wrong!

Press (Y) to continue Quiz If you want to end this Quiz then press (N)

y

2.Q-How many bytes consume Int data type in 64 bit OS?

(A).4 (B).6

(C).2 (D).8

Choose which option is correct:a

Answer is correct!

Press (Y) to continue Quiz If you want to end this Quiz then press (N)

y

3.Q-How many bytes consume Float data type in 64 bit OS?

(A).6 (B).4

(C).2 (D).8

Choose which option is correct:b

Answer is correct!

Press (Y) to continue Quiz If you want to end this Quiz then press (N)

y

4.Q-How many bytes consume Double data type in 64 bit OS?

(A).4 (B).6

(C).2 (D).8

Choose which option is correct:d

Answer is correct!

Press (Y) to continue Quiz If you want to end this Quiz then press (N)

y

5.Q-How many bytes consume char data type in 64 bit OS?

(A).4 (B).6

(C).1 (D).8

Choose which option is correct:c

Answer is correct!

Press (Y) to continue Quiz If you want to end this Quiz then press (N)

y

6.Q-Which type of values is store in Int data type?

(A).integer (B).floating point

(C).character (D).string

Choose which option is correct:a

Answer is correct!

Press (Y) to continue Quiz If you want to end this Quiz then press (N)

y

7.Q-Which type of values is store in float data type?

(A).integer (B).floating point

(C).character (D).string

Choose which option is correct:b

Answer is correct!

Press (Y) to continue Quiz If you want to end this Quiz then press (N)

y

8.Q-Which type of values is store in double data type?

(A).integer (B).floating point

(C).character (D).string

Choose which option is correct:c

Sorry Answer is wrong!

Press (Y) to continue Quiz If you want to end this Quiz then press (N)

y

9.Q-What is string?

(A).Integer values (B).floating values

(C).Character array (D).Array

Choose which option is correct:c

Answer is correct!

Press (Y) to continue Quiz If you want to end this Quiz then press (N)

y

10.Q-What is structure?

(A).Integer (B).float

(C).Character (D).User defined data type

Choose which option is correct:c

Sorry Answer is wrong!

Press (Y) to continue Quiz If you want to end this Quiz then press (N)

y

Congrats! pranjal You are win the quiz.

you got 70 numbers.

your 7 Question in correct.

PS C:\Users\shriv\Downloads\pranjal>

**6.Profiling—**

Profiling code…

1.gcc -Wall -pg miniproject.c -test

2. ./test…will create gmon.out. file

3. gprof test gmon.out > output…. This will convert gmon.out to readable format.

Flat profile:

Each sample counts as 0.01 seconds.

no time accumulated

% cumulative self self total

time seconds seconds calls Ts/call Ts/call name

0.00 0.00 0.00 10 0.00 0.00 result

0.00 0.00 0.00 1 0.00 0.00 calculateScore

0.00 0.00 0.00 1 0.00 0.00 menu

0.00 0.00 0.00 1 0.00 0.00 question

% the percentage of the total running time of the

time program used by this function.

cumulative a running sum of the number of seconds accounted

seconds for by this function and those listed above it.

self the number of seconds accounted for by this

seconds function alone. This is the major sort for this

listing.

calls the number of times this function was invoked, if

this function is profiled, else blank.

self the average number of milliseconds spent in this

ms/call function per call, if this function is profiled,

else blank.

total the average number of milliseconds spent in this

ms/call function and its descendents per call, if this

function is profiled, else blank.

name the name of the function. This is the minor sort

for this listing. The index shows the location of

the function in the gprof listing. If the index is

in parenthesis it shows where it would appear in

the gprof listing if it were to be printed.

Copyright (C) 2012-2017 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,

are permitted in any medium without royalty provided the copyright

notice and this notice are preserved.

Call graph (explanation follows)

granularity: each sample hit covers 4 byte(s) no time propagated

index % time self children called name

0.00 0.00 10/10 question [5]

[2] 0.0 0.00 0.00 10 result [2]

-----------------------------------------------

0.00 0.00 1/1 question [5]

[3] 0.0 0.00 0.00 1 calculateScore [3]

-----------------------------------------------

0.00 0.00 1/1 main [84]

[4] 0.0 0.00 0.00 1 menu [4]

0.00 0.00 1/1 question [5]

-----------------------------------------------

0.00 0.00 1/1 menu [4]

[5] 0.0 0.00 0.00 1 question [5]

0.00 0.00 10/10 result [2]

0.00 0.00 1/1 calculateScore [3]

-----------------------------------------------

This table describes the call tree of the program, and was sorted by

the total amount of time spent in each function and its children.

Each entry in this table consists of several lines. The line with the

index number at the left hand margin lists the current function.

The lines above it list the functions that called this function,

and the lines below it list the functions this one called.

This line lists:

index A unique number given to each element of the table.

Index numbers are sorted numerically.

The index number is printed next to every function name so

it is easier to look up where the function is in the table.

% time This is the percentage of the `total' time that was spent

in this function and its children. Note that due to

different viewpoints, functions excluded by options, etc,

these numbers will NOT add up to 100%.

self This is the total amount of time spent in this function.

children This is the total amount of time propagated into this

function by its children.

called This is the number of times the function was called.

If the function called itself recursively, the number

only includes non-recursive calls, and is followed by

a `+' and the number of recursive calls.

name The name of the current function. The index number is

printed after it. If the function is a member of a

cycle, the cycle number is printed between the

function's name and the index number.

For the function's parents, the fields have the following meanings:

self This is the amount of time that was propagated directly

from the function into this parent.

children This is the amount of time that was propagated from

the function's children into this parent.

called This is the number of times this parent called the

function `/' the total number of times the function

was called. Recursive calls to the function are not

included in the number after the `/'.

name This is the name of the parent. The parent's index

number is printed after it. If the parent is a

member of a cycle, the cycle number is printed between

the name and the index number.

If the parents of the function cannot be determined, the word

`<spontaneous>' is printed in the `name' field, and all the other

fields are blank.

For the function's children, the fields have the following meanings:

self This is the amount of time that was propagated directly

from the child into the function.

children This is the amount of time that was propagated from the

child's children to the function.

called This is the number of times the function called

this child `/' the total number of times the child

was called. Recursive calls by the child are not

listed in the number after the `/'.

name This is the name of the child. The child's index

number is printed after it. If the child is a

member of a cycle, the cycle number is printed

between the name and the index number.

If there are any cycles (circles) in the call graph, there is an

entry for the cycle-as-a-whole. This entry shows who called the

cycle (as parents) and the members of the cycle (as children.)

The `+' recursive calls entry shows the number of function calls that

were internal to the cycle, and the calls entry for each member shows,

for that member, how many times it was called from other members of

the cycle.

Copyright (C) 2012-2017 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,

are permitted in any medium without royalty provided the copyright

notice and this notice are preserved.

**Index by function name**

**[3] calculateScore [5] question**

**[4] menu [2] result**

**6.Debugging—**

Debugging code…

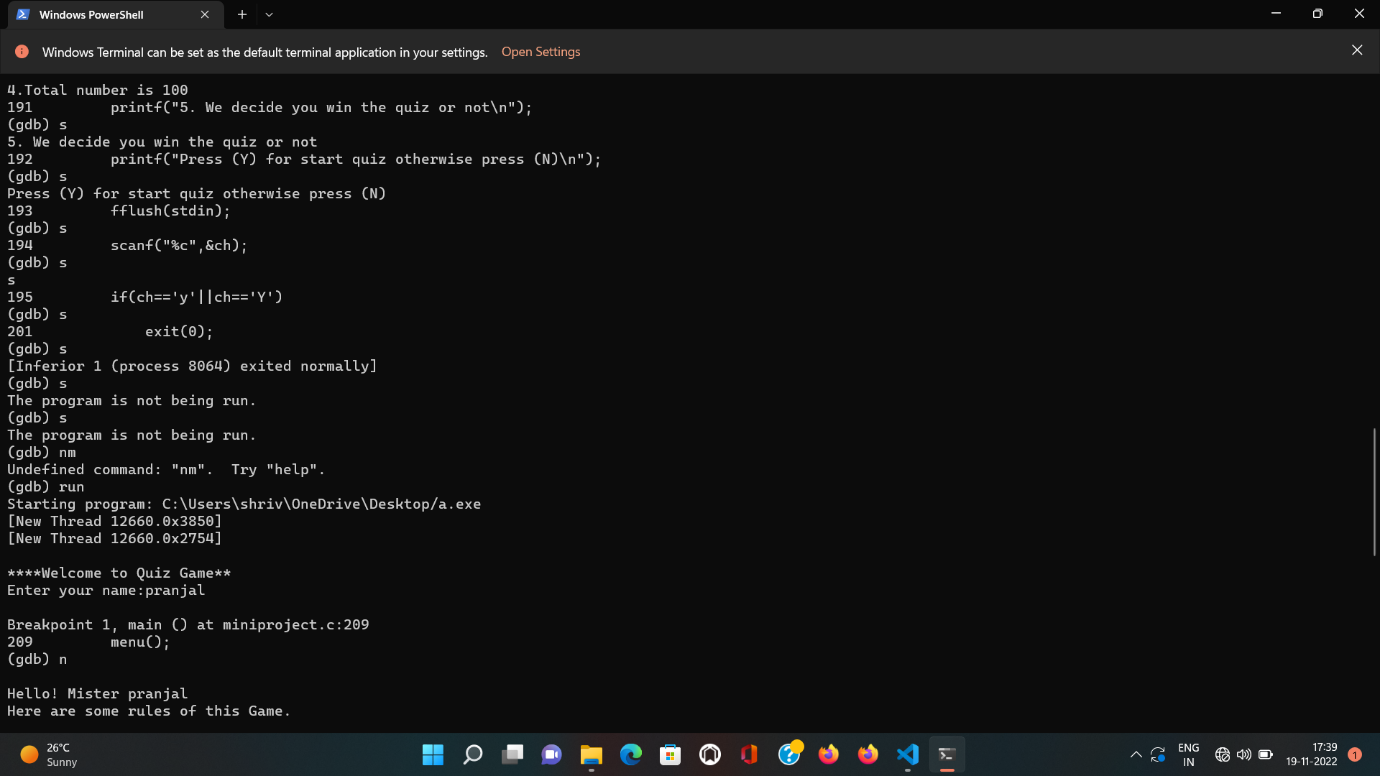
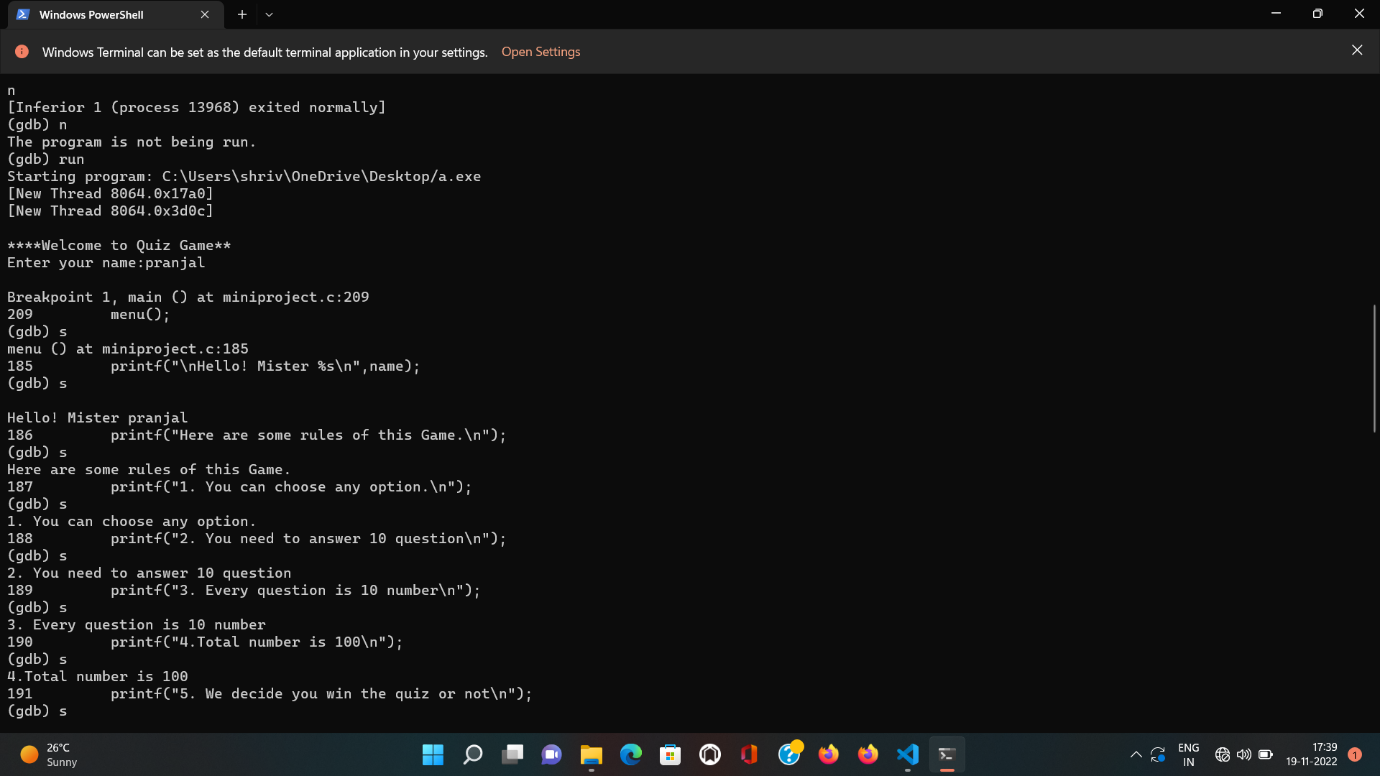
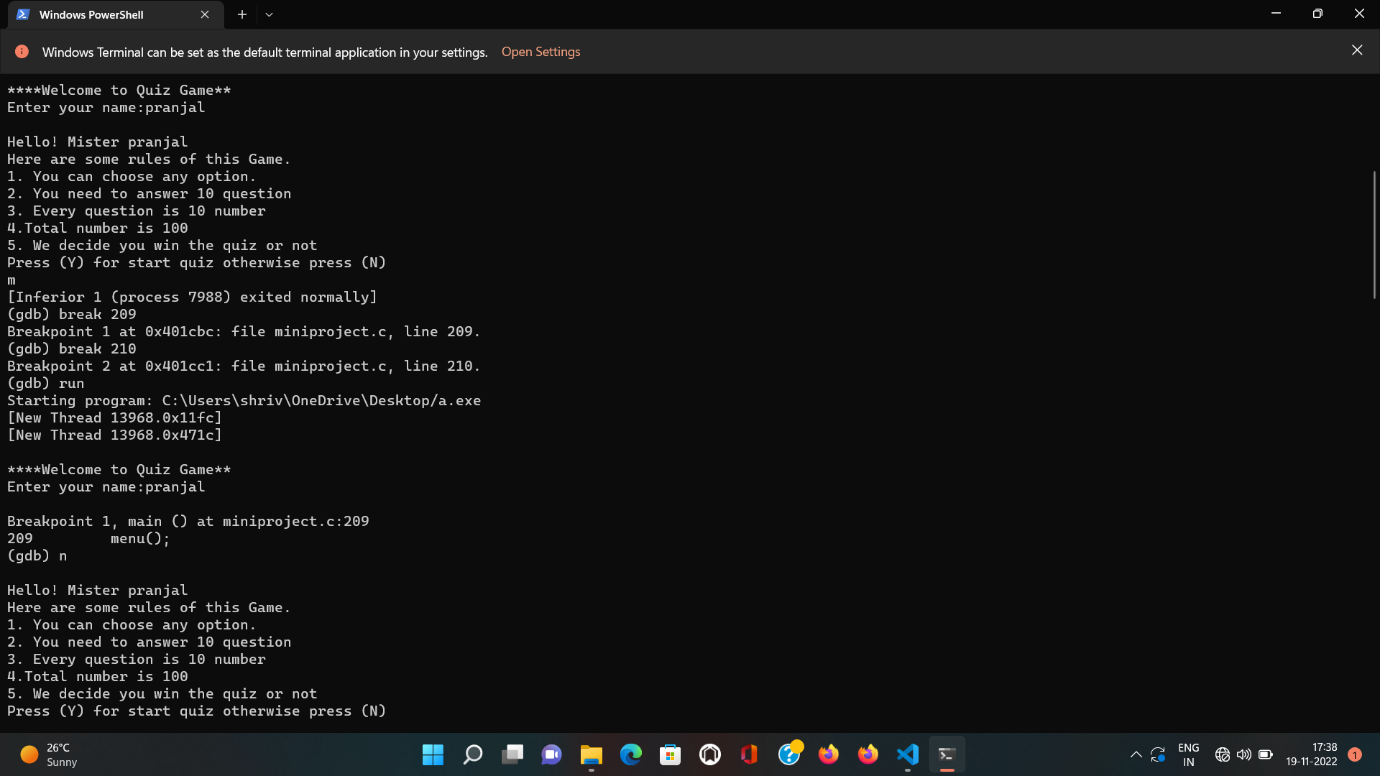
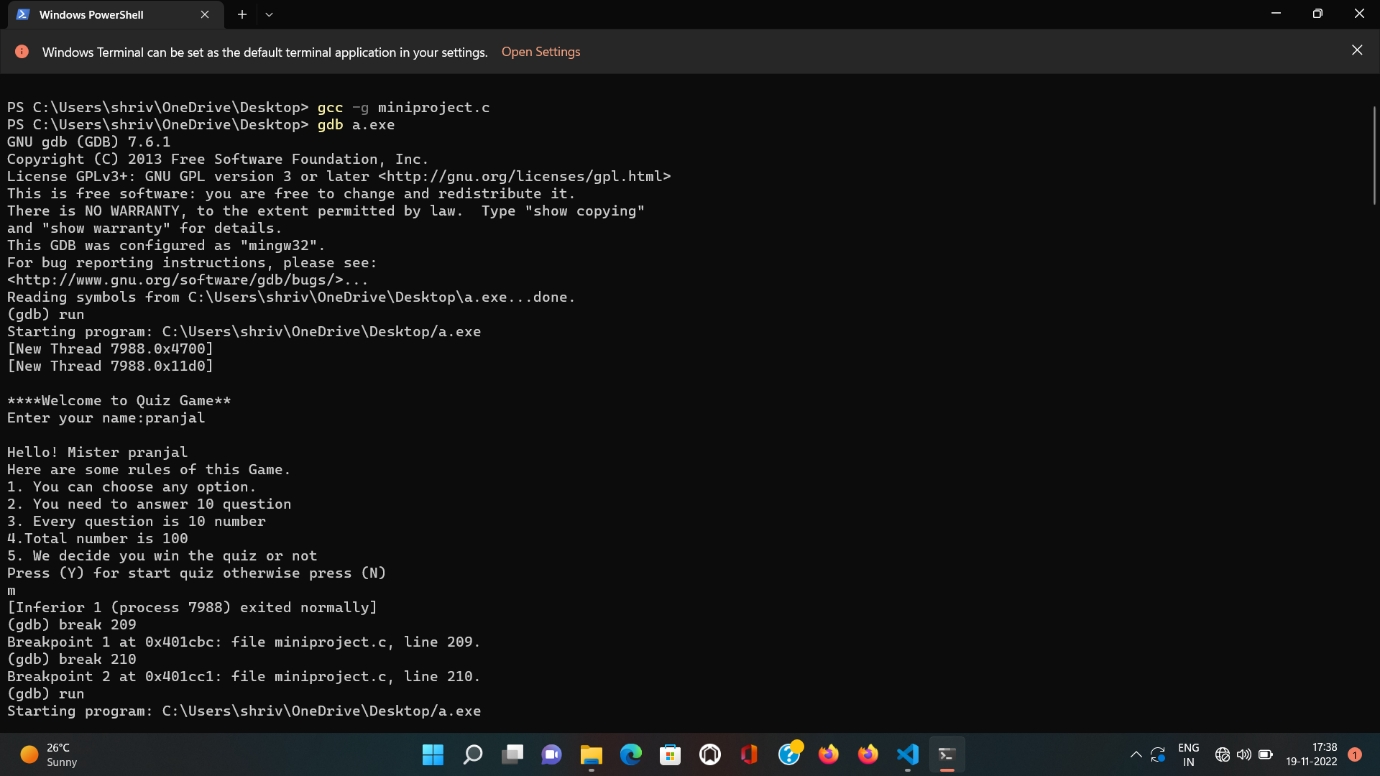
1.gcc -g miniproject.c

2.gdb a.exe…gdb will start

3.break linenumber

4.run

5. q or quit



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*THE END\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*