**Roll No: 1803067**

**Lab Performance Test 1**

**Lab Task Q1 , Q2a , Q2b**

**Question:**

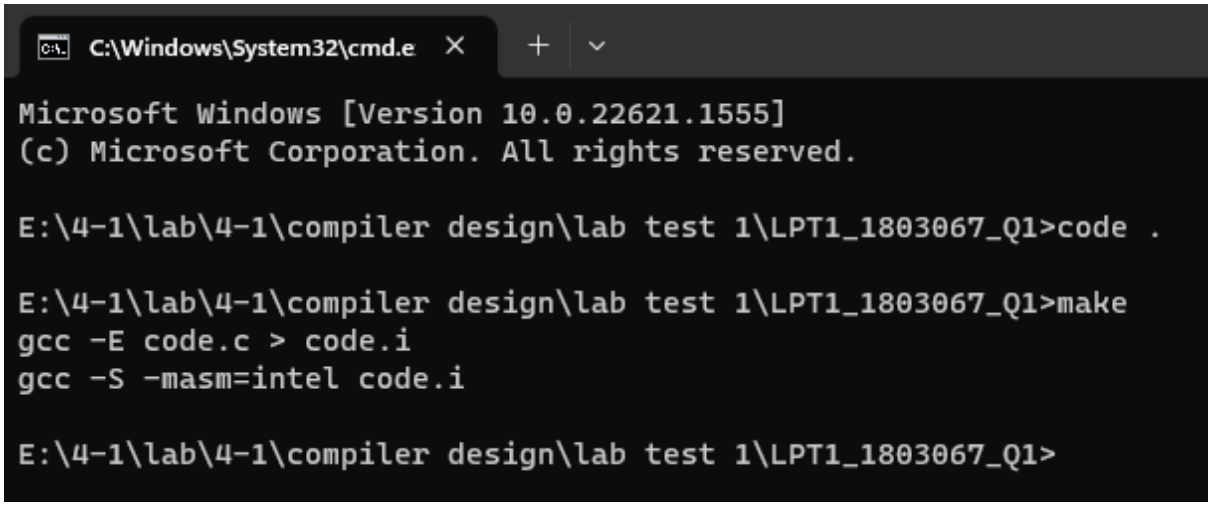
Q1. Consider given code, #include<math.h> int main(){ float a=1; float b=2; float c=a+b; return 0; } . Show output file with extension ".s" generated by C compiler along with Makefile (point penalty for adding extra commands other than necessary).

Q2. Consider given statements, 1. We bought 1 Apple 2. You sold 50 Mangoes 3. We Wanted 0 Oranges. a) Show a flex file which can tokenize given statements. b) Show a bison file which can parse given statements.

**Q1. Solution (Bold your own written code):**

```
all:
    gcc -E code.c > code.i
    gcc -S -masm=intel code.i
```

**Output (Screen/SnapShot):**




**Q2A. Solution (Bold your own written code):**

```
.l
%option noyywrap
```

```
%{


%}


%%
"1"|"50"|"0"    {printf("ADJ =>%s\t",yytext);}
"We"|"You"    {printf("N =>%s\t",yytext);}
"bought"|"sold"|"Wanted"    {printf("V
=>%s\t",yytext);}
"Apple"|"Mangoes"|"Oranges" {printf("O
=>%s\t",yytext);}
%%

int main()
{
    yylex();
    return 0;
}
```

Makefile

```
main:
    flex flex.l
    gcc lex.yy.c
    a.exe <input.txt
```

**Output (Screen/SnapShot):**

```
E:\4-1\lab\4-1\compiler design\lab test 1\LPT1_1803067_Q2a>make
flex flex.l
gcc lex.yy.c
a.exe <input.txt
N =>We    V =>bought      ADJ =>1         O =>Apple
N =>You   V =>sold        ADJ =>50        O =>Mangoes
N =>We    V =>Wanted      ADJ =>0         O =>Oranges
E:\4-1\lab\4-1\compiler design\lab test 1\LPT1_1803067_Q2a>S
```

**Q1. Solution (Bold your own written code):**

.l file

```
%option noyywrap

%{
    //roll : 1803067
#include "bison.tab.h"
%}



%%
"1"|"50"|"0"    {return(ADJ);}
"We"|"You"    {return(N);}
"bought"|"sold"|"Wanted"    {return(V);}
"Apple"|"Mangoes"|"Oranges" {return(O);}
. {}
%%
```

.y file

```
%{
#include<stdio.h>
void yyerror(char *s);
int yylex();
%}
```

```
%token N V O ADJ
%start S

%%
S: S A | A
A: N V O | N V ADJ O;
%%

int f=1;

int main()
{
      //roll : 1803067

    yyparse();
    if(f)
    {
        printf("Accepted\n");
    }
    return 0;
}

void yyerror(char *s)
{
    f=0;
    fprintf(stderr, "<---- error: %s\n", s);
}
```

Makefile

```
main:
    bison -d bison.y
    flex flex.l
    gcc bison.tab.c lex.yy.c
    a.exe <input.txt
```

**Output (Screen/SnapShot):**

```
E:\4-1\lab\4-1\compiler design\lab test 1\LPT1_1803067_Q2b>make
bison -d bison.y
flex flex.l
gcc bison.tab.c lex.yy.c
a.exe <input.txt


Accepted
```