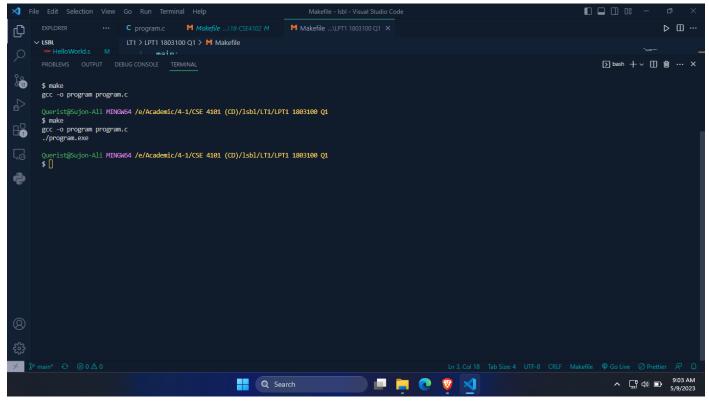## Question 1:- Q1. Consider given code, #include<math.h> int main(){ float a=10.1; float b=20.2; float c=a+b; return 0; } . Show assembly output file with filename "assemblyprogram.txt" generated by C compiler along with Makefile (point penalty for adding extra commands other than necessary).

Code:- Make File Code

```
main:
    gcc -o program program.c
    ./program.exe
```

Output:-

Question 2a:- Consider given statements, 1.He Loves Apple 2. He Dislikes Pizza 3. He Likes Parata.
a) Show a flex file which can tokenize given statements.

Code:-

Flex File Code:

```
%option noyywrap

%{
#include "program2.tab.h"
%}

%%
```

```
"He" {printf("%s --> Pronoun \n",yytext); return(PORNOUN);}
"Loves" {printf("%s --> Verb1 \n",yytext); return(VERB1);}
"Dislikes" {printf("%s --> Verb2 \n",yytext); return(VERB2);}
"Likes" {printf("%s --> Verb3 \n",yytext); return(VERB3);}
"Pizza" {printf("%s --> Object1 \n",yytext); return(OBJECT1);}
"Apple" {printf("%s --> Object2 \n",yytext); return(OBJECT2);}
"Parata" {printf("%s --> Object3 \n",yytext); return(OBJECT3);}
%%
```

Output:-

Question 2b:- Consider given statements, 1.He Loves Apple 2. He Dislikes Pizza 3. He Likes Parata.
b) Show a bison file which can parse given statements.

Code:-

Bison File Code :

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

%}

%token PORNOUN   VERB1 VERB2 VERB3 OBJECT1 OBJECT2 OBJECT3
%start s

%%
s: PORNOUN   VERB1 VERB2 VERB3 OBJECT1 OBJECT2 OBJECT3
    ;
%%


int main(){
    yyparse();
    printf("\nparsing finished");
    return 0;
}

void yyerror(char *s){
    fprintf(stderr,"\nError : %s",s);
}
```

Flex File Code:

```
%option noyywrap

%{
#include "program2.tab.h"
%}

%%
"He" {printf("%s --> Pronoun \n",yytext); return(PORNOUN);}
"Loves" {printf("%s --> Verb1 \n",yytext); return(VERB1);}
"Dislikes" {printf("%s --> Verb2 \n",yytext); return(VERB2);}
"Likes" {printf("%s --> Verb3 \n",yytext); return(VERB3);}
"Pizza" {printf("%s --> Object1 \n",yytext); return(OBJECT1);}
"Apple" {printf("%s --> Object2 \n",yytext); return(OBJECT2);}
"Parata" {printf("%s --> Object3 \n",yytext); return(OBJECT3);}
%%
```

Output:-

```
flex program2.l
gcc program2.tab.c lex.yy.c
./a.exe < input.txt
He --> Pronoun
 Loves --> Verb1
 Dislikes --> Verb2
 Likes --> Verb3
 Pizza --> Object1
 Apple --> Object2
 Parata --> Object3

 parsing finished
```