	AIR UNIVERSITY
	DEPARTMENT OF COMPUTER SCIENCE
	Lab Task 3

Student Name: Hamza Umer Farooq **Reg. No: 200789**

Subject: Compiler Construction **Semester: VIII**

Objective: Tokenization

ASSESSMENT:

Attributes	Excellent (5)	Good (4)	Average (3)	Satisfactory (2)	Unsatisfactory (1)
Ability to Conduct Task					
Ability to assimilate the results					
Effective use of theorems/postulates/formulas					

Total Marks:
Obtained Marks:

REPORT ASSESSMENT:

Attributes	Excellent (5)	Good (4)	Average (3)	Satisfactory (2)	Unsatisfactory (1)
Data presentation					
Experimental results					
Conclusion					

Code to check:

```
if (a >= 10)
{ Sum = x;
print (“value is x”);
}
else{
return 30;
}
```

Macros:

```
%{
#include <stdio.h>
%}

%option noyywrap


DIGIT [0-9]
LETTER [a-zA-Z]
WHITESPACE [ \t\n]+
%%
{DIGIT}+ {printf("Integer: %s\n", yytext);}
{LETTER}+ {printf("Identifier: %s\n", yytext);}
{WHITESPACE} {}
. {printf("Unknown Token: %s\n",yytext);}
%%

int main() {
//
}
```

Lexemes/Tokens according to rules/macros defined above:

<Identifier: if>
<Unknown Token: (>
<Identifier: a>
<Unknown Token: >>
<Unknown Token: =>
<Integer: 10>
<Unknown Token:)>
<Unknown Token: {>

```
<Identifier: Sum>
<Unknown Token: =>
<Identifier: x>
<Unknown Token: ;>
<Unknown Token: }>
<Identifier: else>
<Unknown Token: {>
<Identifier: return>
<Integer: 30>
<Unknown Token: ;>
<Unknown Token: }>
```

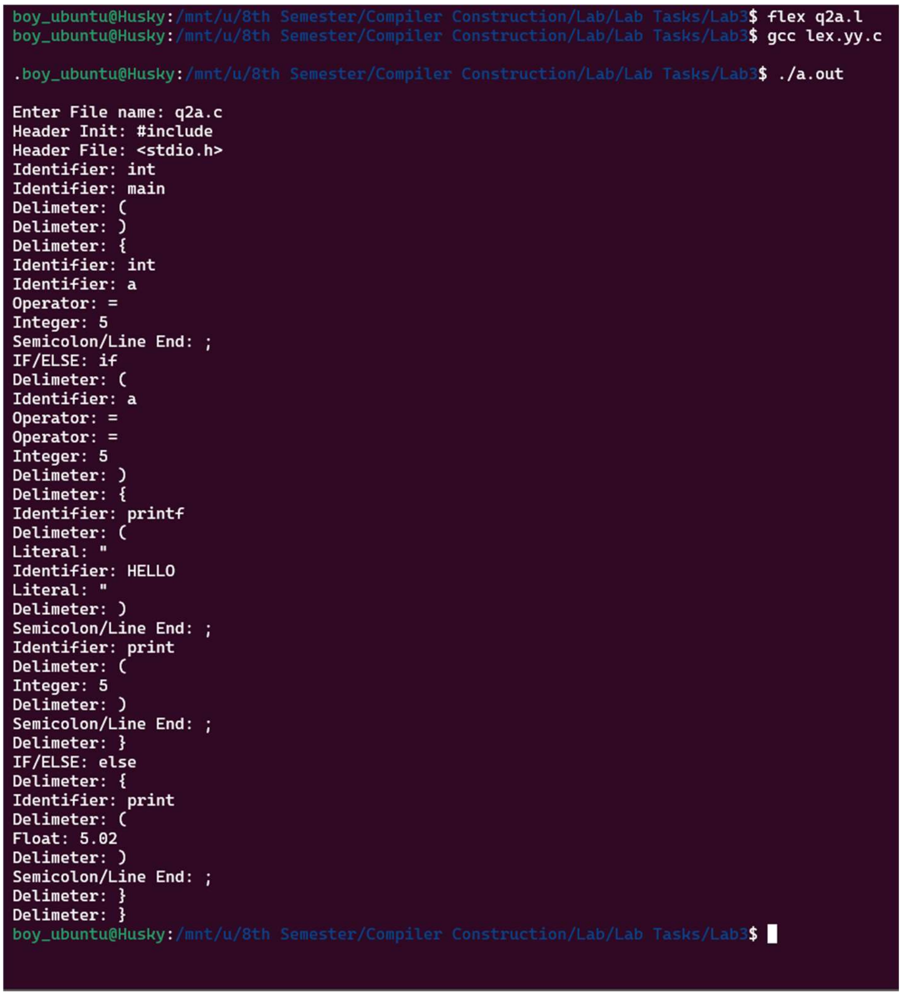
QUESTION 2A

IF ELSE

```
%{
#include <stdio.h>
%}

%option noyywrap


DIGIT [0-9]
IF_ELSE if|else
FLOAT (0|[1-9][0-9]*)\.[0-9]+
INT {DIGIT}+
HEADER_INIT #include
HEADER_FILE <[a-zA-Z0-9_]+\.h>


%%
{HEADER_INIT} { printf("Header Init: %s\n",yytext);}
{HEADER_FILE} { printf("Header File: %s\n", yytext); }
{IF_ELSE} { printf("IF/ELSE: %s\n",yytext);}
{INT} { printf("Integer: %s\n", yytext); }
{FLOAT} { printf("Float: %s\n", yytext); }
[a-zA-Z][a-zA-Z0-9]* { printf("Identifier: %s\n", yytext); }
[(){}] {printf("Delimeter: %s\n",yytext);}
[;] {printf("Semicolon/Line End: %s\n",yytext);}
[:] {printf("Colon: %s\n",yytext);}
[,] {printf("Comma: %s\n",yytext);}
"[" {printf("Square Bracket: %s\n",yytext);}
"]" {printf("Square Bracket: %s\n",yytext);}
[+\-*/] { printf("Arith Operator: %s\n", yytext); }
[=+><] { printf("Operator: %s\n",yytext);}
```

```
[\".*\"] {printf("Literal: %s\n",yytext);}
[\t\n] {}
" " {}
. { printf("Unknown Token: %s\n",yytext);}
%%

int main() {
    FILE* fp;
    char filename[30];
    printf("\nEnter File name: ");
    scanf("%s",filename);
    fp = fopen(filename,"r");
    yyin = fp;
    yylex();
}
```

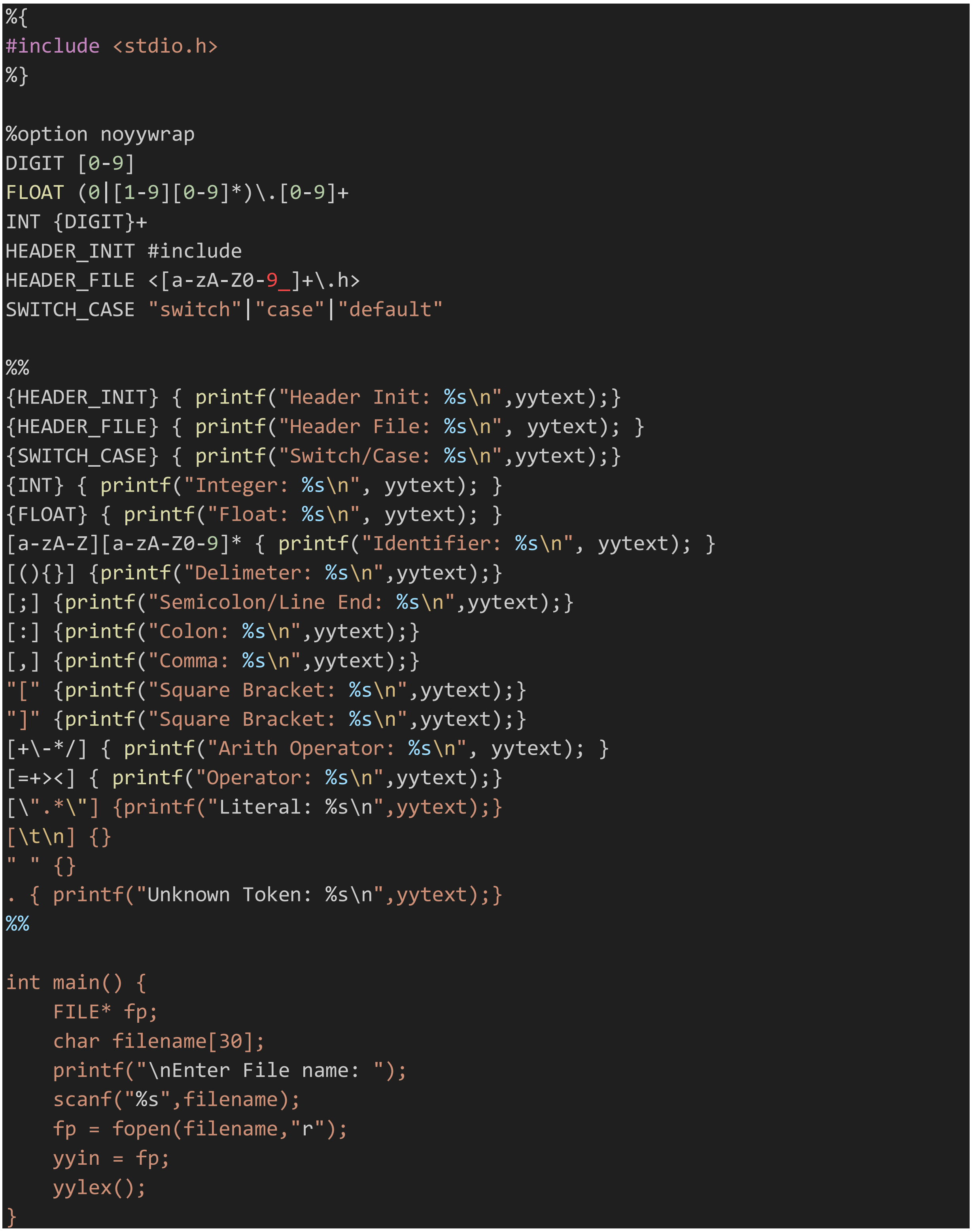
OUTPUT:



```
boy_ubuntu@Husky:/mnt/u/8th Semester/Compiler Construction/Lab/Lab Tasks/Lab3$ flex q2a.l
boy_ubuntu@Husky:/mnt/u/8th Semester/Compiler Construction/Lab/Lab Tasks/Lab3$ gcc lex.yy.c

.boy_ubuntu@Husky:/mnt/u/8th Semester/Compiler Construction/Lab/Lab Tasks/Lab3$ ./a.out

Enter File name: q2a.c
Header Init: #include
Header File: <stdio.h>
Identifier: int
Identifier: main
Delimeter: (
Delimeter: )
Delimeter: {
Identifier: int
Identifier: a
Operator: =
Integer: 5
Semicolon/Line End: ;
IF/ELSE: if
Delimeter: (
Identifier: a
Operator: =
Operator: =
Integer: 5
Delimeter: )
Delimeter: {
Identifier: printf
Delimeter: (
Literal: "
Identifier: HELLO
Literal: "
Delimeter: )
Semicolon/Line End: ;
Identifier: print
Delimeter: (
Integer: 5
Delimeter: )
Semicolon/Line End: ;
Delimeter: }
IF/ELSE: else
Delimeter: {
Identifier: print
Delimeter: (
Float: 5.02
Delimeter: )
Semicolon/Line End: ;
Delimeter: }
Delimeter: }
boy_ubuntu@Husky:/mnt/u/8th Semester/Compiler Construction/Lab/Lab Tasks/Lab3$
```

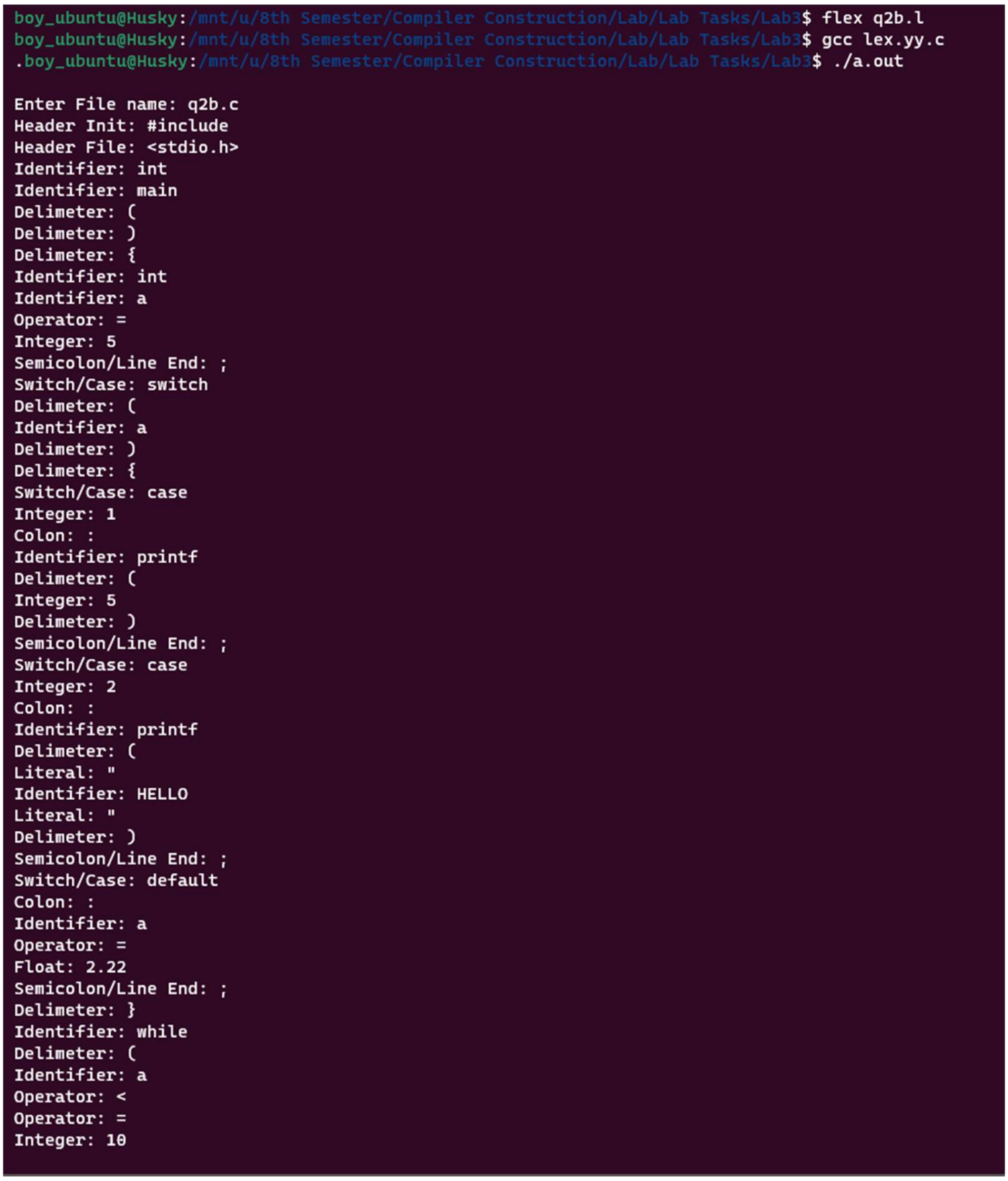
QUESTION 2B

SWITCH-CASE



```
%{
#include <stdio.h>
%}

%option noyywrap
DIGIT [0-9]
FLOAT (0|[1-9][0-9]*)\.[0-9]+
INT {DIGIT}+
HEADER_INIT #include
HEADER_FILE <[a-zA-Z0-9_]+\.h>
SWITCH_CASE "switch"|"case"|"default"

%%
{HEADER_INIT} { printf("Header Init: %s\n",yytext);}
{HEADER_FILE} { printf("Header File: %s\n", yytext); }
{SWITCH_CASE} { printf("Switch/Case: %s\n",yytext);}
{INT} { printf("Integer: %s\n", yytext); }
{FLOAT} { printf("Float: %s\n", yytext); }
[a-zA-Z][a-zA-Z0-9]* { printf("Identifier: %s\n", yytext); }
[(){}] {printf("Delimeter: %s\n",yytext);}
[;] {printf("Semicolon/Line End: %s\n",yytext);}
[:] {printf("Colon: %s\n",yytext);}
[,] {printf("Comma: %s\n",yytext);}
"[" {printf("Square Bracket: %s\n",yytext);}
"]" {printf("Square Bracket: %s\n",yytext);}
[+\-*/] { printf("Arith Operator: %s\n", yytext); }
[=+><] { printf("Operator: %s\n",yytext);}
[\".*\"] {printf("Literal: %s\n",yytext);}
[\t\n] {}
" " {}
. { printf("Unknown Token: %s\n",yytext);}
%%

int main() {
    FILE* fp;
    char filename[30];
    printf("\nEnter File name: ");
    scanf("%s",filename);
    fp = fopen(filename,"r");
    yyin = fp;
    yylex();
}
```

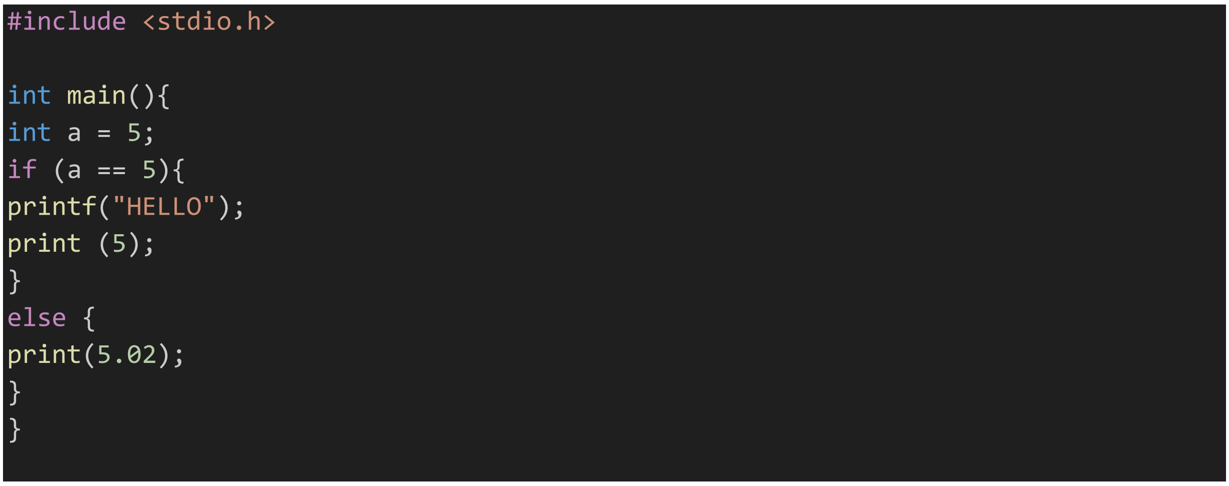
OUTPUT



```
boy_ubuntu@Husky:/mnt/u/8th Semester/Compiler Construction/Lab/Lab Tasks/Lab3$ flex q2b.l
boy_ubuntu@Husky:/mnt/u/8th Semester/Compiler Construction/Lab/Lab Tasks/Lab3$ gcc lex.yy.c
.boy_ubuntu@Husky:/mnt/u/8th Semester/Compiler Construction/Lab/Lab Tasks/Lab3$ ./a.out

Enter File name: q2b.c
Header Init: #include
Header File: <stdio.h>
Identifier: int
Identifier: main
Delimeter: (
Delimeter: )
Delimeter: {
Identifier: int
Identifier: a
Operator: =
Integer: 5
Semicolon/Line End: ;
Switch/Case: switch
Delimeter: (
Identifier: a
Delimeter: )
Delimeter: {
Switch/Case: case
Integer: 1
Colon: :
Identifier: printf
Delimeter: (
Integer: 5
Delimeter: )
Semicolon/Line End: ;
Switch/Case: case
Integer: 2
Colon: :
Identifier: printf
Delimeter: (
Literal: "
Identifier: HELLO
Literal: "
Delimeter: )
Semicolon/Line End: ;
Switch/Case: default
Colon: :
Identifier: a
Operator: =
Float: 2.22
Semicolon/Line End: ;
Delimeter: }
Identifier: while
Delimeter: (
Identifier: a
Operator: <
Operator: =
Integer: 10
```

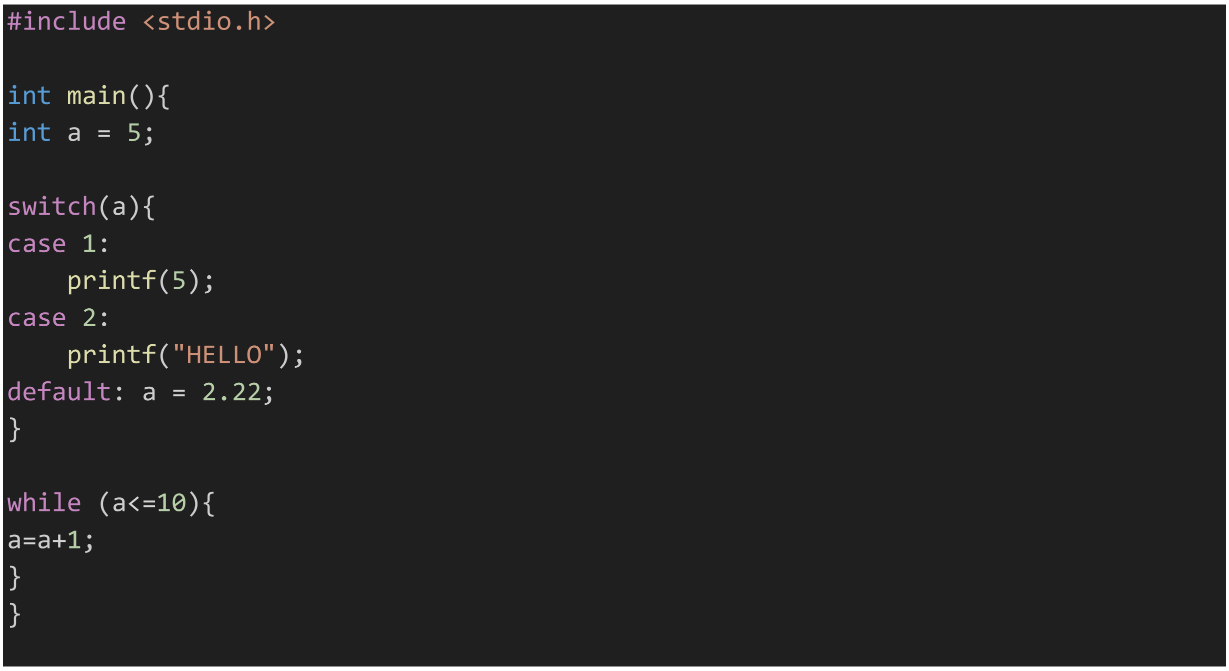
Sample codes used for testing q2a.l and q2b.l

q2a.c



```c
#include <stdio.h>

int main(){
int a = 5;
if (a == 5){
printf("HELLO");
print (5);
}
else {
print(5.02);
}
}
```

q2b.c



```c
#include <stdio.h>

int main(){
int a = 5;

switch(a){
case 1:
    printf(5);
case 2:
    printf("HELLO");
default: a = 2.22;
}

while (a<=10){
a=a+1;
}
}
```