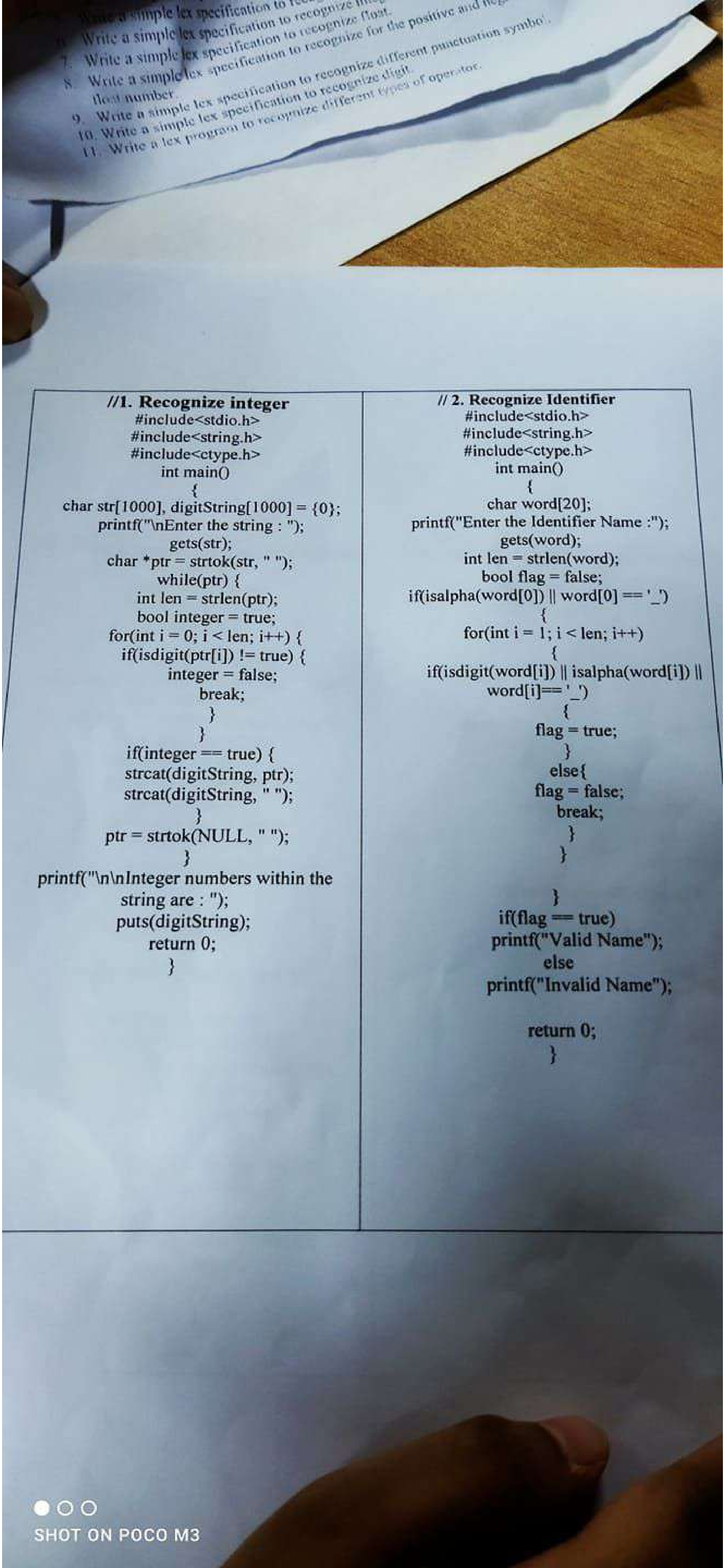


5. Write a simple lex specification to reco

6. Write a simple lex specification to recognize inte

7. Write a simple lex specification to recognize float.

8. Write a simple lex specification to recognize for the positive and neg float number.

9. Write a simple lex specification to recognize different punctuation symbol.

10. Write a simple lex specification to recognize digit.

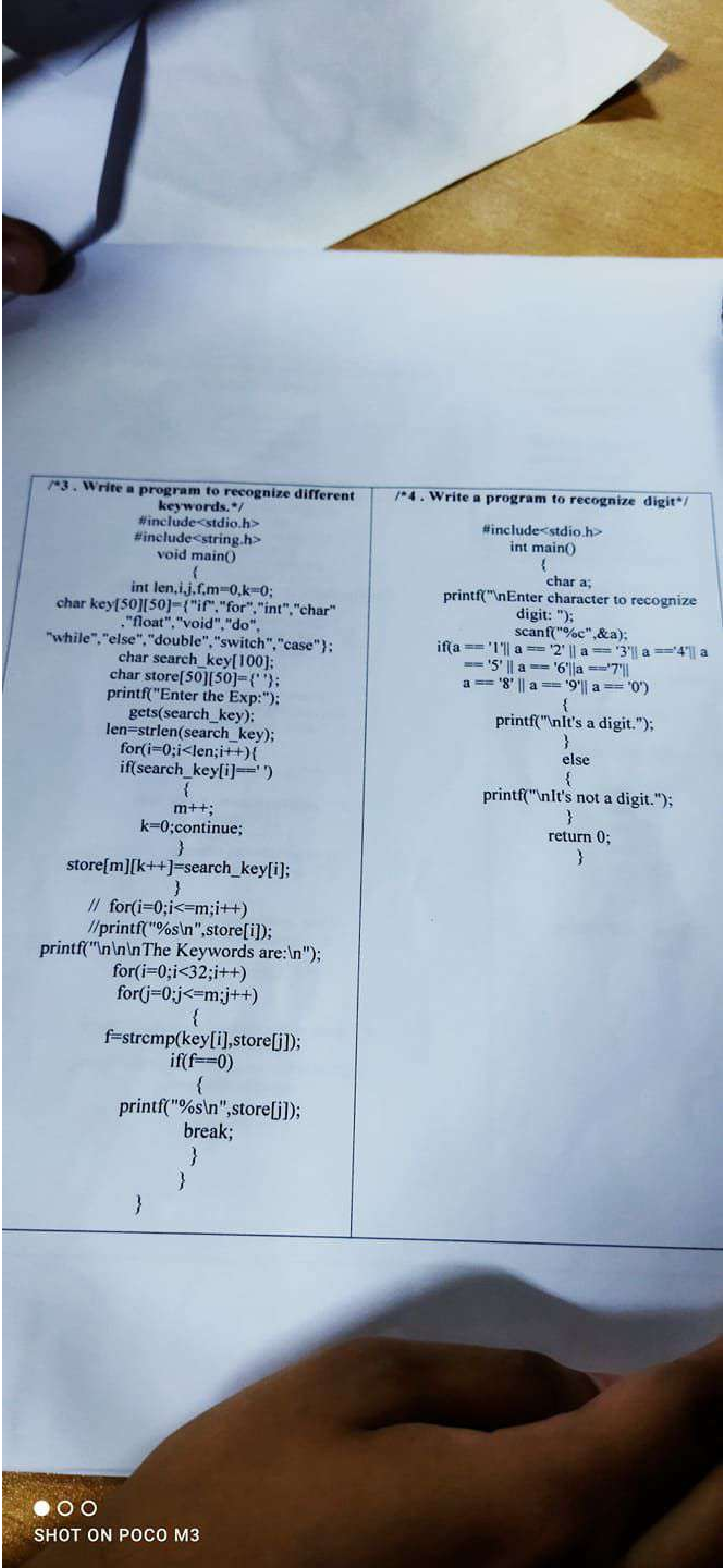
11. Write a lex program to recognize different types of operator.

//1. Recognize integer

```
#include<stdio.h>
#include<string.h>
#include<ctype.h>
int main()
{
char str[1000], digitString[1000] = {0};
printf("\nEnter the string : ");
gets(str);
char *ptr = strtok(str, " ");
while(ptr) {
int len = strlen(ptr);
bool integer = true;
for(int i = 0; i < len; i++) {
if(isdigit(ptr[i]) != true) {
integer = false;
break;
}
}
if(integer == true) {
strcat(digitString, ptr);
strcat(digitString, " ");
}
ptr = strtok(NULL, " ");
}
printf("\n\nInteger numbers within the
string are : ");
puts(digitString);
return 0;
}
```

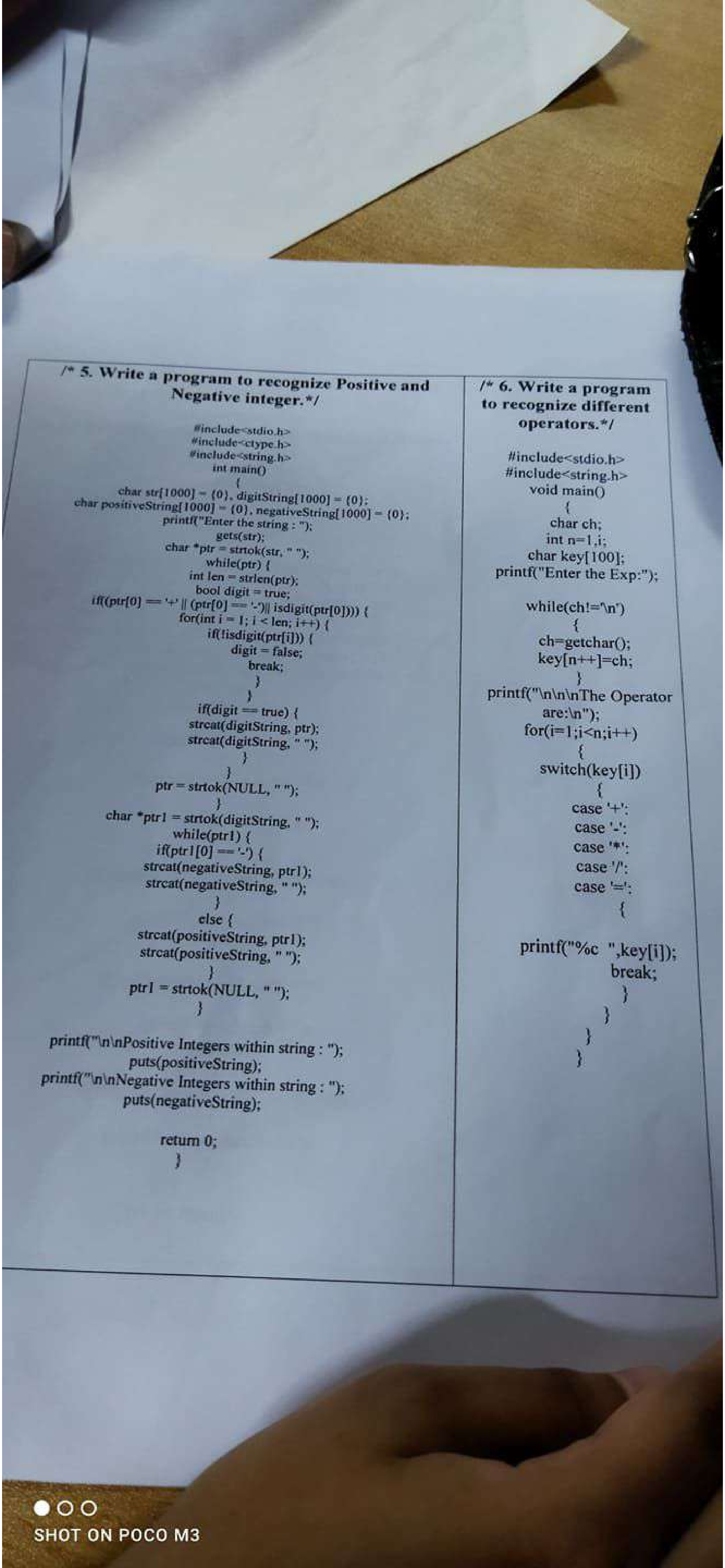
// 2. Recognize Identifier

```
#include<stdio.h>
#include<string.h>
#include<ctype.h>
int main()
{
char word[20];
printf("Enter the Identifier Name :");
gets(word);
int len = strlen(word);
bool flag = false;
if(isalpha(word[0]) || word[0] == '_')
{
for(int i = 1; i < len; i++)
{
if(isdigit(word[i]) || isalpha(word[i]) ||
word[i]== '_')
{
flag = true;
}
else{
flag = false;
break;
}
}

}
if(flag == true)
printf("Valid Name");
else
printf("Invalid Name");

return 0;
}
```



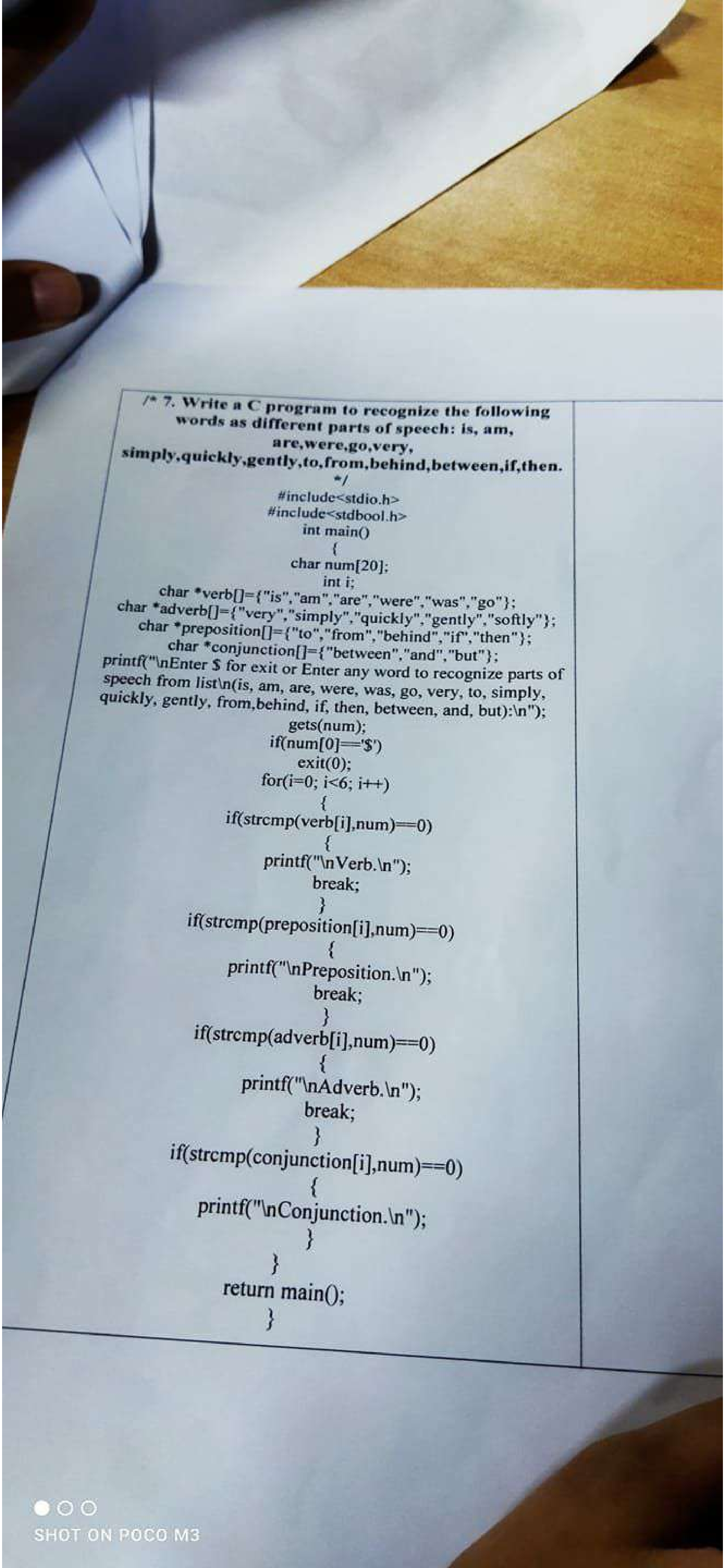
```
/*3 . Write a program to recognize different
keywords.*/
#include<stdio.h>
#include<string.h>
void main()
{
int len,i,j,f,m=0,k=0;
char key[50][50]={"if","for","int","char"
,"float","void","do",
"while","else","double","switch","case"};
char search_key[100];
char store[50][50]={' '};
printf("Enter the Exp:");
gets(search_key);
len=strlen(search_key);
for(i=0;i<len;i++){
if(search_key[i]==' ')
{
m++;
k=0;continue;
}
store[m][k++]=search_key[i];
}
// for(i=0;i<=m;i++)
//printf("%s\n",store[i]);
printf("\n\n\nThe Keywords are:\n");
for(i=0;i<32;i++)
for(j=0;j<=m;j++)
{
f=strcmp(key[i],store[j]);
if(f==0)
{
printf("%s\n",store[j]);
break;
}
}
}
```

```
/*4 . Write a program to recognize digit*/

#include<stdio.h>
int main()
{
char a;
printf("\nEnter character to recognize
digit: ");
scanf("%c",&a);
if(a == '1'|| a == '2' || a == '3'|| a =='4'|| a
== '5' || a == '6'||a =='7'||
a == '8' || a == '9'|| a == '0')
{
printf("\nIt's a digit.");
}
else
{
printf("\nIt's not a digit.");
}
return 0;
}
```



```
/* 5. Write a program to recognize Positive and
Negative integer.*/

#include<stdio.h>
#include<ctype.h>
#include<string.h>
int main()
{
char str[1000] = {0}, digitString[1000] = {0};
char positiveString[1000] = {0}, negativeString[1000] = {0};
printf("Enter the string : ");
gets(str);
char *ptr = strtok(str, " ");
while(ptr) {
int len = strlen(ptr);
bool digit = true;
if((ptr[0] == '+' || (ptr[0] == '-')|| isdigit(ptr[0]))) {
for(int i = 1; i < len; i++) {
if(!isdigit(ptr[i])) {
digit = false;
break;
}
}
if(digit == true) {
strcat(digitString, ptr);
strcat(digitString, " ");
}
}
ptr = strtok(NULL, " ");
}
char *ptr1 = strtok(digitString, " ");
while(ptr1) {
if(ptr1[0] == '-') {
strcat(negativeString, ptr1);
strcat(negativeString, " ");
}
else {
strcat(positiveString, ptr1);
strcat(positiveString, " ");
}
ptr1 = strtok(NULL, " ");
}

printf("\n\nPositive Integers within string : ");
puts(positiveString);
printf("\n\nNegative Integers within string : ");
puts(negativeString);

return 0;
}
```

```
/* 6. Write a program
to recognize different
operators.*/

#include<stdio.h>
#include<string.h>
void main()
{
char ch;
int n=1,i;
char key[100];
printf("Enter the Exp:");

while(ch!='\n')
{
ch=getchar();
key[n++]=ch;
}
printf("\n\n\nThe Operator
are:\n");
for(i=1;i<n;i++)
{
switch(key[i])
{
case '+':
case '-':
case '*':
case '/':
case '=':
{

printf("%c  ",key[i]);
break;
}
}
}
}
```



```
/* 7. Write a C program to recognize the following
words as different parts of speech: is, am,
are,were,go,very,
simply,quickly,gently,to,from,behind,between,if,then.
*/
#include<stdio.h>
#include<stdbool.h>
int main()
{
char num[20];
int i;
char *verb[]={"is","am","are","were","was","go"};
char *adverb[]={"very","simply","quickly","gently","softly"};
char *preposition[]={"to","from","behind","if","then"};
char *conjunction[]={"between","and","but"};
printf("\nEnter $ for exit or Enter any word to recognize parts of
speech from list\n(is, am, are, were, was, go, very, to, simply,
quickly, gently, from,behind, if, then, between, and, but):\n");
gets(num);
if(num[0]=='$')
exit(0);
for(i=0; i<6; i++)
{
if(strcmp(verb[i],num)==0)
{
printf("\nVerb.\n");
break;
}
if(strcmp(preposition[i],num)==0)
{
printf("\nPreposition.\n");
break;
}
if(strcmp(adverb[i],num)==0)
{
printf("\nAdverb.\n");
break;
}
if(strcmp(conjunction[i],num)==0)
{
printf("\nConjunction.\n");
}
}
return main();
}
```