EXPERIMENT NO: 03

Title: Program to create a symbol table generator
Aim: Program to create a symbol table generator

Theory:

A Symbol table is a data structure used by a language translator such as a compiler or Interpreter, where each identifier in a program's source code is associated with information Relating to its declaration or appearance in the source

Possible entries in a symbol table:

- Name: a string
- Attribute:

1. Reserved word
2. Variable name
3. Type Name
4. Procedure name
5. Constant name

- Data type
- Scope information: where it can be used.
- Storage allocation

ALGORITHM:

1. Start the Program.
2. Get the input from the user with the terminating symbol '$'.
3. Allocate memory for the variable by dynamic memory allocation function.
4. If the next character of the symbol is an operator then only the memory is allocated.
5. While reading, the input symbol is inserted into symbol table along with its memory address.
6. The steps are repeated till "$"is reached.
7. To reach a variable, enter the variable to the searched and symbol table has been checked for corresponding variable, the variable along its address is displayed as result.
8. Stop the program.

PROGRAM: (IMPLEMENTATION OF SYMBOL TABLE)

```
#include<stdio.h>
#include<conio.h>
#include<malloc.h>
#include<string.h>
#include<math.h>
```

```
#include<ctype.h>
void main()
{
int i=0,j=0,x=0,n,flag=0; void *p,*add[15];
char ch,srch,b[15],d[15],c;
//clrscr();
printf("expression terminated by $:");
while((c=getchar())!='$')
{
b[i]=c; i++;
}
n=i-1;
printf("given expression:");
i=0;
while(i<=n)
{
printf("%c",b[i]); i++;
}
printf("symbol table\n");
printf("symbol\taddr\ttype\n");
while(j<=n)
{
c=b[j]; if(isalpha(toascii(c)))
{
if(j==n)
{
p=malloc(c); add[x]=p;
d[x]=c;
printf("%c\t%d\tidentifier\n",c,p);
}
else
{
ch=b[j+1];
if(ch=='+'||ch=='-'||ch=='*'||ch=='=')
{
p=malloc(c);
add[x]=p;
d[x]=c;
printf("%c\t%d\tidentifier\n",c,p);
x++;
```

```
}
}
} j++;
}
printf("the symbol is to be searched\n");
srch=getch();
for(i=0;i<=x;i++)
{
if(srch==d[i])
{
printf("symbol found\n");
printf("%c%s%d\n",srch,"@address",add[i]);
flag=1;
}
}
if(flag==0)
printf("symbol not found\n");
//getch();
```

Sample Questions:

1. **What is Symbol Table Generator?**
2. **List Various Attributes of Symbol Table.**
3. **Explain Working of Symbol table attributes.**