**PRACTICAL – 9**

|  |  |
| --- | --- |
|  | **Problem statement**  **Write a program which generates Quadruple Table for the given post fix String** |

**SOLUTION:**

|  |
| --- |
| **CODE:**  #include <stdio.h>  #include <stdlib.h>  #include <string.h>  #include <ctype.h>  #define MAX 100  typedef struct {  char op[3];  char arg1[10];  char arg2[10];  char result[10];  } Quadruple;  Quadruple quadruples[MAX];  int quadIndex = 0;  void generateQuadruple(char op[], char arg1[], char arg2[], char result[]) {  strcpy(quadruples[quadIndex].op, op);  strcpy(quadruples[quadIndex].arg1, arg1);  strcpy(quadruples[quadIndex].arg2, arg2);  strcpy(quadruples[quadIndex].result, result);  quadIndex++;  }  void printQuadruples() {  printf("Quadruple Table:\n");  printf("Op\tArg1\tArg2\tResult\n");  for (int i = 0; i < quadIndex; i++) {  printf("%s\t%s\t%s\t%s\n", quadruples[i].op, quadruples[i].arg1, quadruples[i].arg2, quadruples[i].result);  }  }  void generateQuadrupleTable(char postfix[]) {  char stack[MAX][10];  int top = -1;  char temp[2] = "t";  int tempIndex = 1;  for (int i = 0; i < strlen(postfix); i++) {  if (isalnum(postfix[i])) {  char operand[2] = {postfix[i], '\0'};  strcpy(stack[++top], operand);  } else {  char arg2[10], arg1[10], result[10];  strcpy(arg2, stack[top--]);  strcpy(arg1, stack[top--]);  sprintf(result, "t%d", tempIndex++);  char op[2] = {postfix[i], '\0'};  generateQuadruple(op, arg1, arg2, result);  strcpy(stack[++top], result);  }  }  }  int main() {  char postfix[MAX];  printf("Enter a postfix expression: ");  scanf("%s", postfix);  generateQuadrupleTable(postfix);  printQuadruples();  return 0;  }  **OUTPUT:** |