14. INTERMEDIATE CODE: QUADRUPLE

CODE:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct Quadruple {

char op[3];

char arg1[10];

char arg2[10];

char result[10];

};

struct Quadruple quadruples[100];

int numQuadruples = 0,i;

void addQuadruple(const char \*op, const char \*arg1, const char \*arg2, const char \*result) {

strcpy(quadruples[numQuadruples].op, op);

strcpy(quadruples[numQuadruples].arg1, arg1);

strcpy(quadruples[numQuadruples].arg2, arg2);

strcpy(quadruples[numQuadruples].result, result);

numQuadruples++;

}

int main() {

// Example: Generate quadruples for expression: a = b + c \* d

addQuadruple("+", "b", "c", "t1");

addQuadruple("\*", "t1", "d", "t2");

addQuadruple("=", "t2", "", "a");

printf("Generated Quadruples:\n");

printf("operator ,arg1,arg2,result:\n");

for (i = 0; i < numQuadruples; i++) {

printf("(%s , %s, %s, %s)\n",

quadruples[i].op,

quadruples[i].arg1,

quadruples[i].arg2,

quadruples[i].result);

}

return 0;

}

OUTPUT:

Generated Quadruples:

operator ,arg1,arg2,result:

(+ , b, c, t1)

(\* , t1, d, t2)

(= , t2, , a)

--------------------------------

Process exited after 0.06647 seconds with return value 0

Press any key to continue . . .

