**ASSEMBLER CODE (C++)**

#include<stdio.h>#include<iostream>#include<string>#include<fstream>#include<sstream>using namespace std;int binary(int);int main(){ // assembley language instruction string add = "ADD"; string sub = "SUB"; string orr = "OR"; string ornot = "ORNOT"; string andd = "AND"; string andnot = "ANDNOT"; string compacc = "COMPACC"; string comp = "COMP"; string load = "LOAD"; string str = "STR"; string shiftr = "SHIFTR"; string shiftl = "SHIFTL"; string mul = "MUL"; string div = "DIV"; string branch = "BRANCH"; string brzero = "BRZERO"; string jumpsub = "JUMPSUB"; string returnsub = "RETURNSUB"; string returnitr = "REUTRNITR"; string input = "INPUT"; string output = "OUTPUT"; string maskhvpi = "MASKHVPI"; // Addressing modes string immediate = "imm"; string direct = "dir"; string indirect = "indir"; ifstream file("assembly.txt"); string line; if (!file) { cout <<"unable to open file"<<endl; } while(getline(file,line)) { istringstream iss(line); string instruct; string adrmode; string num; string instBin; string adrBin; getline(iss,instruct,' '); getline(iss,adrmode,' '); getline(iss,num); if(instruct == add) { instBin = "00000"; } if(instruct == sub) { instBin = "00001"; } if(instruct == orr) { instBin = "00010"; } if(instruct == ornot) { instBin = "00011"; } if(instruct == andd) { instBin = "00100"; } if(instruct == andnot) { instBin = "00101"; }if(instruct == sub) { instBin = "00001"; } if(instruct == orr) { instBin = "00010"; } if(instruct == ornot) { instBin = "00011"; } if(instruct == andd) { instBin = "00100"; } if(instruct == andnot) { instBin = "00101"; } if(instruct == compacc) { instBin = "00110"; } if(instruct == comp) { instBin = "00111"; } if(instruct == load) { instBin = "01000"; } if(instruct == str) { instBin = "01001"; } if(instruct == shiftl) { instBin = "01010"; } if(instruct == shiftr) { instBin = "01011"; } if(instruct == input) { instBin = "01100"; } if(instruct == output) { instBin = "01101"; } if(instruct == maskhvpi) { instBin = "01111"; } if(instruct == mul) { instBin = "11000"; } if(instruct == div) { instBin = "11001"; } if(instruct == branch) { instBin = "11010"; } if(instruct == brzero) { instBin = "11011"; } if(instruct == jumpsub) { instBin = "11100"; } if(instruct == returnsub) { instBin = "11101"; } if(instruct == returnitr) { instBin = "11110"; } if(adrmode == immediate) { adrBin = "000"; } if(adrmode == direct) { adrBin = "001"; } if(adrmode == indirect) { adrBin = "010"; } int dec = atoi(num.c\_str()); int rem,i=1,sum=0; do { rem=dec%2; sum=sum + (i\*rem); dec=dec/2; i=i\*10; }while(dec>0); cout << instBin<<" "<<adrBin<<" "<<sum<<endl; } file.close(); return 0;}