

THỰC HÀNH VI XỬ LÝ – VI ĐIỀU KHIỂN

GVHD: Bùi Phùng Hữu Đức

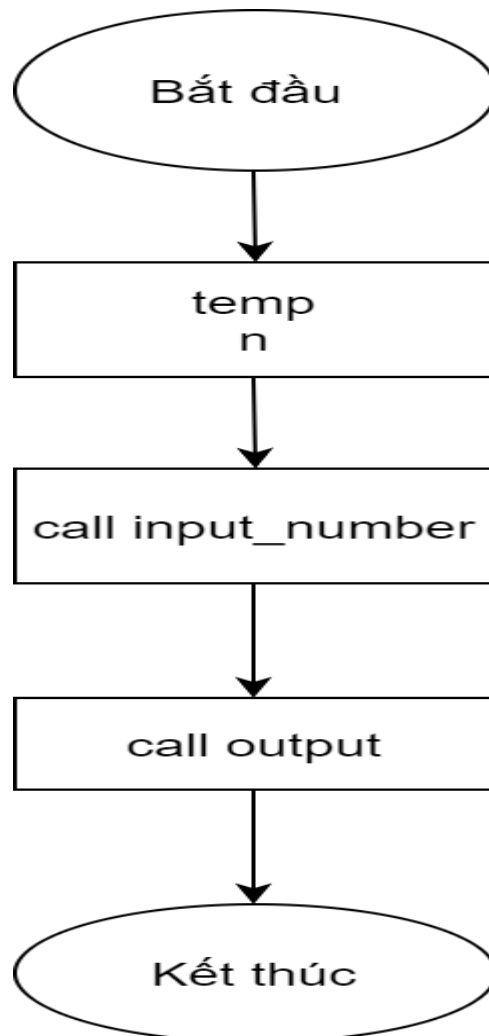
Họ và tên sinh viên thực hiện: Nguyễn Hữu Tứ

Mã số sinh viên: 19522453

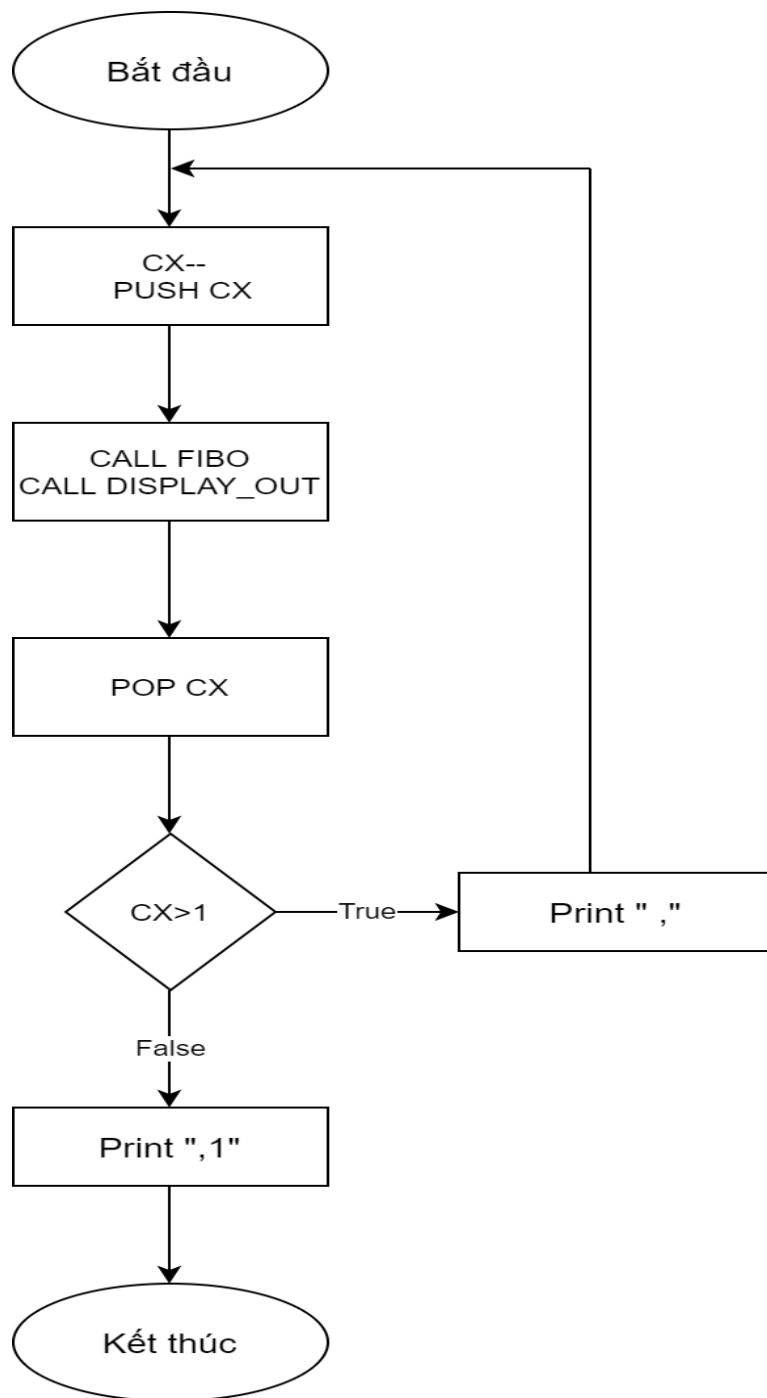
---

BÁO CÁO THỰC HÀNH SỐ 06  
XỬ LÝ IO, TÍNH TOÁN VÀ BỘ NHỚ TRÊN 8086

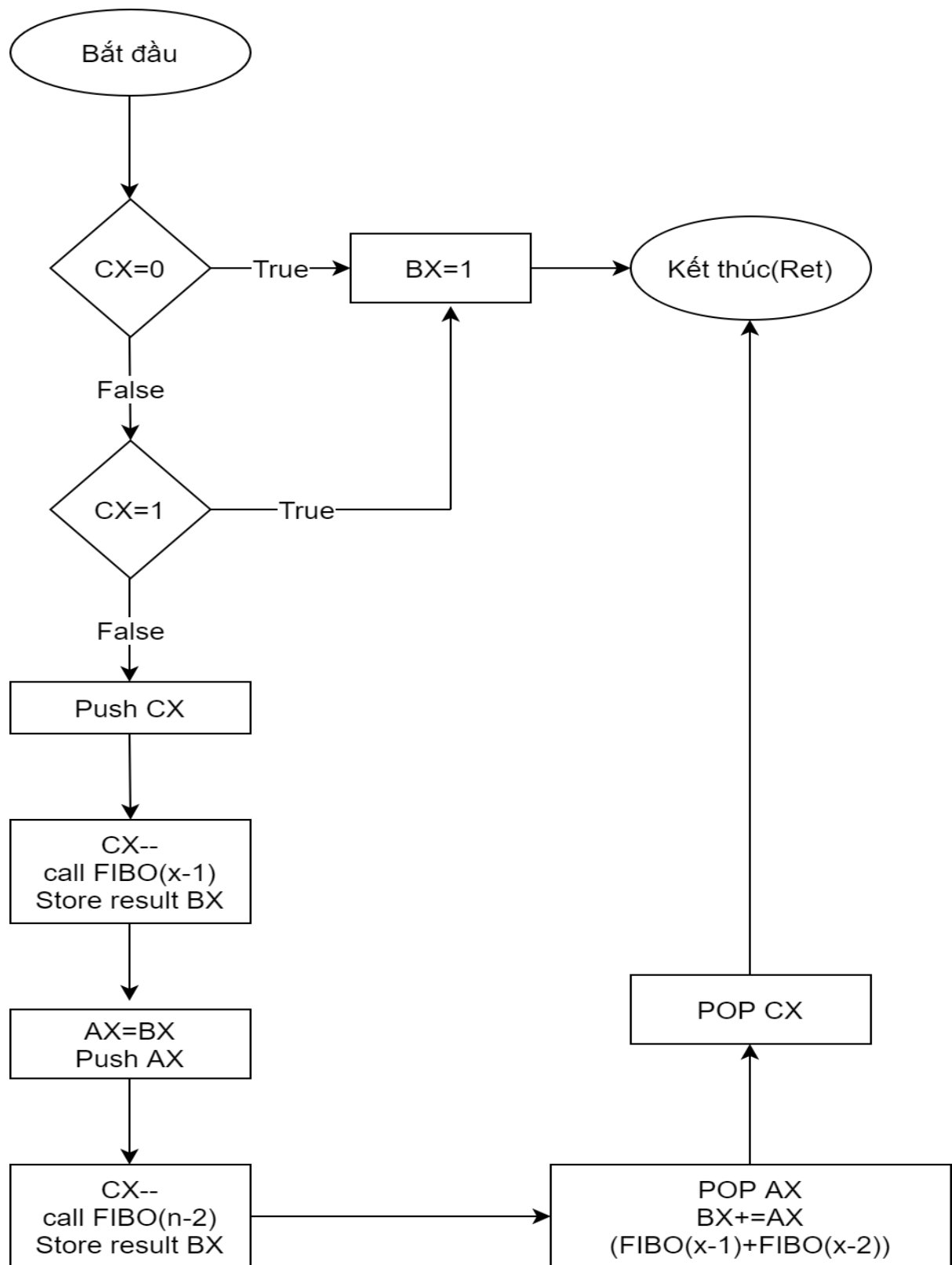
- I. Nhập vào một số nguyên N và in ra N số fibonacci đầu tiên :
1. Lưu đồ thuật toán



Hình 1 : Lưu đồ hàm main



Hình 2 :Sơ đồ hàm Output



Hình 3 :Sơ đồ hàm FIBO

Code	Giải thích
.MODEL SMALL .STACK 100H	

<pre> .DATA temp DW ? tb1 db "Nhap N :\$" under_line DB 0AH,0DH,"\$" .CODE MAIN PROC     MOV AX,@DATA     MOV DS,AX      MOV AH,9     LEA DX,tb1     INT 21h     MOV AH,01H     CALL INPUT_NUMBER     MOV AH,9     LEA DX,under_line     INT 21h     DB 13,10,0     CALL OUTPUT  MAIN ENDP ;///////////////////////////////// DISPLAY_ PROC MOV CS:temp, SI POP SI PUSH AX nextchar: MOV AL, CS:[SI] INC SI CMP AL, 0 JZ print MOV AH, 0Eh INT 10h JMP nextchar print: POP AX PUSH SI MOV SI, CS:temp RET DISPLAY_ ENDP ;///////////////////////////////// INPUT_NUMBER PROC MOV BX,0h MOV CX, 0h next_digit: MOV AH, 00h </pre>	<p>Biến temp lưu giá trị tạm với không có giá trị khởi tạo.</p> <p>Chuỗi tb1 thông báo nhập n</p> <p>Chuỗi under_line dùng để xuống dòng</p> <p>.Phần code</p> <p>Hàm main</p> <p>Dùng hàm ngắt để xuất ra chuỗi thông báo nhập n</p> <p>Hàm nhập số cần tính toán</p> <p>In ra kí tự xuống dòng để xuất ra chuỗi các số</p> <p>Ta dùng dùng ngắt</p> <p>Hàm output xuất giá trị và tính toán</p> <p>Kết thúc hàm main</p> <p>Hàm dùng để display</p> <p>Gán temp = SI</p> <p>Lấy giá trị vào SI</p> <p>Đưa giá trị AX vào stack</p> <p>Gán AL = temp[SI]</p> <p>Tăng SI (tăng địa chỉ)</p> <p>So sánh AL với 0</p> <p>Nhảy nếu ZF = 1</p> <p>Gán AH = 0Eh để ngắt</p> <p>Nhảy tới nextchar</p> <p>Kết thúc hàm</p> <p>Hàm INPUT_NUM</p> <p>BX = 0</p> <p>CX = 0</p> <p>AH = 0</p>
--	--

<pre> INT 16h MOV AH, 0Eh INT 10h CMP AL, 13 JNE next JMP STOP_INPUT_NUMBER next: MOV BL,AL cmp BX,'0' JB not_minus CMP BX, '9' JA not_minus SUB BX,30h MOV AX, CX MOV CX, 0AH MUL CX MOV CX, AX ADD CX, BX JMP next_digit not_minus: MOV AH,4CH INT 21h RET STOP_INPUT_NUMBER: RET INPUT_NUMBER ENDP  ;///////////////////////////////// FIBO PROC CMP CX,0H JE OUT_FIBO CMP CX,01H JE OUT_FIBO PUSH CX DEC CX CALL FIBO MOV AX,BX DEC CX PUSH AX CALL FIBO POP AX ADD BX,AX POP CX RET OUT_FIBO: MOV BX,01H </pre>	<pre> Interrupt  Interrupt So sánh AL = 13 (nhấn Enter) Nhảy nếu không bằng Nhảy về stop_input_num  BL = AL So sánh BX &gt;= '0'(So sánh ascii) Nhảy nếu điều kiện không đúng So sánh BX &lt;= '9'(So sánh ascii) Nhảy nếu điều kiện trên không đúng BX - 30h (Chuyển về kí tự ascii) AX = CX CX = 10 AX*CX (= AX * 10) CX = AX CX + BX Nhảy  Interrupt terminal  Trở về từ lệnh call  Hàm tính FIBO So sánh CX = 0 Bằng thì nhảy OUT_FIBO So sánh CX = 1 Bằng thì nhảy OUT_FIBO Lưu CX CX -= 1 Gọi FIBO (Fibo(CX - 1)) AX = BX (Hàm trên kết quả lưu BX) CX -= 1 Lưu AX Gọi FIBO(Fibo(CX - 2)) Lấy AX BX = AX+BX= ( Fibo(CX-1) + Fibo(CX-2) ) Trở về từ hàm gọi  BX = 1 </pre>
--	---

<pre> RET FIBO ENDP ;/////////////////////////////////  DISPLAY_OUT PROC     CMP BX, 0AH     JNB OUTPUT_NEXT     ADD BX,30h     MOV DX,BX     MOV AH,02H     INT 21H     RET OUTPUT_NEXT:     MOV CX, 0H     MOV AX, BX DISPLAY:     MOV BX, 10D     MOV DX, 00H     DIV BX     ADD DX, 30H     PUSH DX     INC CX     CMP AX, 0H     JE END_DISPLAY     JMP DISPLAY  END_DISPLAY:     POP DX     MOV AH, 02H     INT 21H     LOOP END_DISPLAY     RET DISPLAY_OUT ENDP ;///////////////////////////////// OUTPUT PROC      DEC CX     PUSH CX     CALL FIBO     CALL DISPLAY_OUT     POP CX     CMP CX,01H     JA OUT_SP     JMP PRINT_1 OUT_SP:     CALL DISPLAY_ </pre>	<p>Trở về từ hàm gọi</p> <p>Kết thúc hàm</p> <p>Hàm DISPLAY_OUT</p> <p>So sánh <math>BX &gt; 10</math></p> <p>Nhảy nếu điều kiện trên sai</p> <p><math>BX + 30h</math></p> <p><math>DX = BX</math></p> <p>Interrupt In kí tự</p> <p>Trở về từ lệnh gọi</p> <p><math>CX = 0</math></p> <p><math>AX = BX</math></p> <p><math>BX = 10</math></p> <p><math>DX = 0</math></p> <p><math>AX = AX/BX</math></p> <p><math>DX + 30</math> (<math>DX = AX \% BX</math>)</p> <p>Lưu DX</p> <p><math>CX + 1</math></p> <p>So sánh <math>AX = 0</math></p> <p>Nhảy nếu bằng</p> <p>Nhảy DISPLAY</p> <p>Lấy DX</p> <p>Interrupt In kí tự</p> <p>LOOP</p> <p>Trở về từ lệnh CALL</p> <p>Kết thúc hàm</p> <p>Giảm CX</p> <p>Lưu CX</p> <p>Gọi hàm FIBO(tính số fibo)</p> <p>Gọi DISPLAY_OUT</p> <p>So sánh <math>CX &gt; 1</math></p> <p>Không đúng thì nhảy OUT_SP (in dấu ,)</p> <p>Nhảy PRINT_1</p> <p>Hàm in ra màn hình</p>
--	---

<pre> DB ", ",0 JMP OUTPUT PRINT_1: CALL DISPLAY_ DB ", 1",0 MOV     AH, 4CH INT     21H  OUTPUT ENDP ;//////////////////////////////// END MAIN </pre>	<p>Nhảy về OUTPUT(lặp lại với giá trị CX giảm dần)</p> <p>Hàm in ra màn hình</p> <p>In kí tự 1 (Chuỗi fibo đã in xong)</p> <p>Ngắt Terminal</p> <p>Kết thúc hàm</p> <p>Kết thúc chương trình</p>
---	--

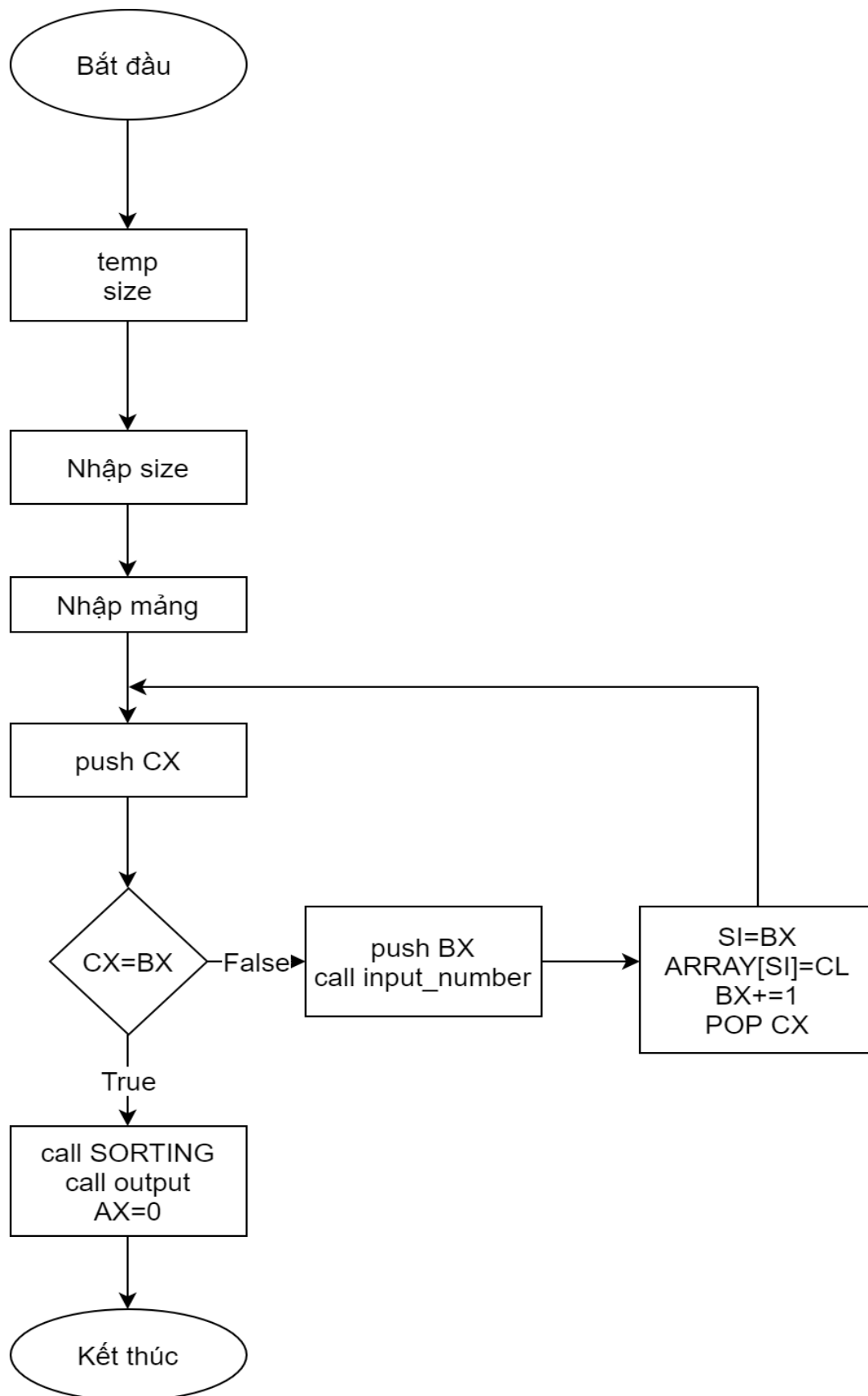
-Kết quả mô phỏng:



Hình 4 : Kết quả mô phỏng

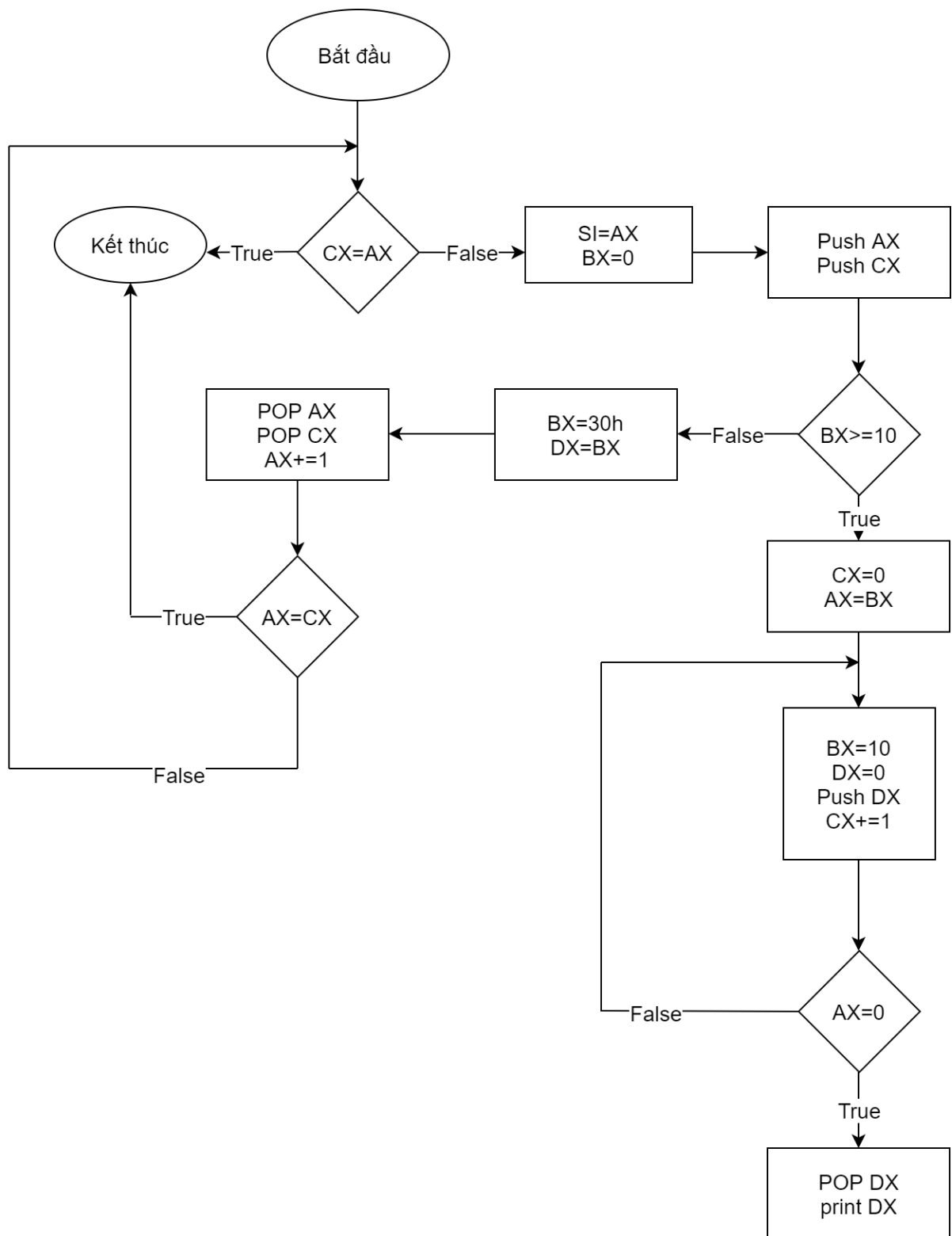
II. Nhập mảng N( $N < 15$ ), sắp xếp theo thứ tự tăng dần và in ra màn hình console:

1. Sơ đồ giải thuật :

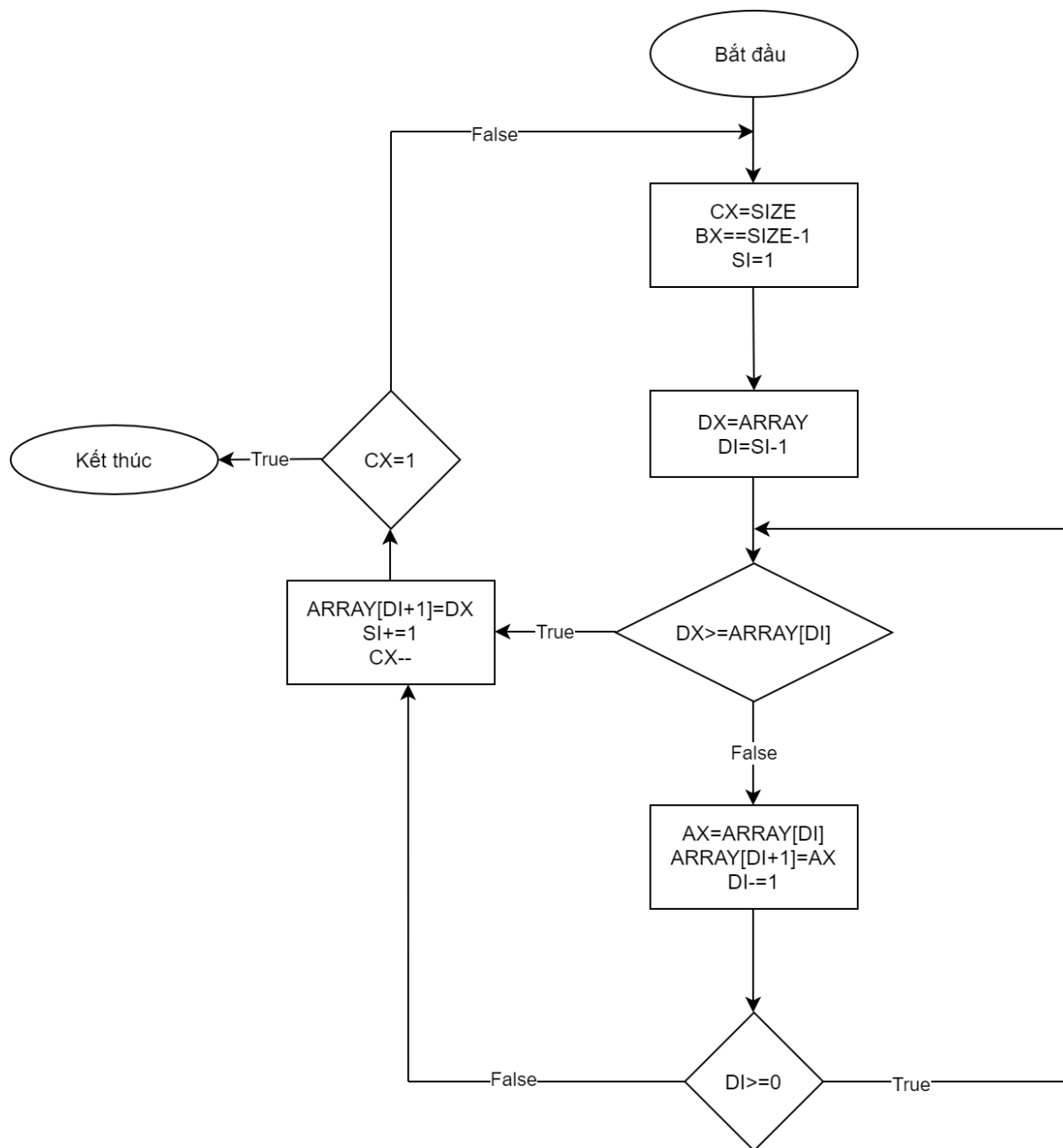


Hình 5 : Sơ đồ hàm main





Hình 6 : Sơ đồ hàm output



Hình 7 : Sơ đồ hàm Sort

Code	Ý nghĩa
<pre> .MODEL SMALL org 100h .STACK 100H .DATA ARRAY DB 15 DUP(?) tb1 db "Nhap so luong phan tu cua mang: \$" tb2 db "Nhap phan tu cua mang: \$" tb3 db "Mang sau khi duoc sap xep la: \$" </pre>	<p>Tạo mảng</p> <p>Các thông báo báo nhập xuất</p>

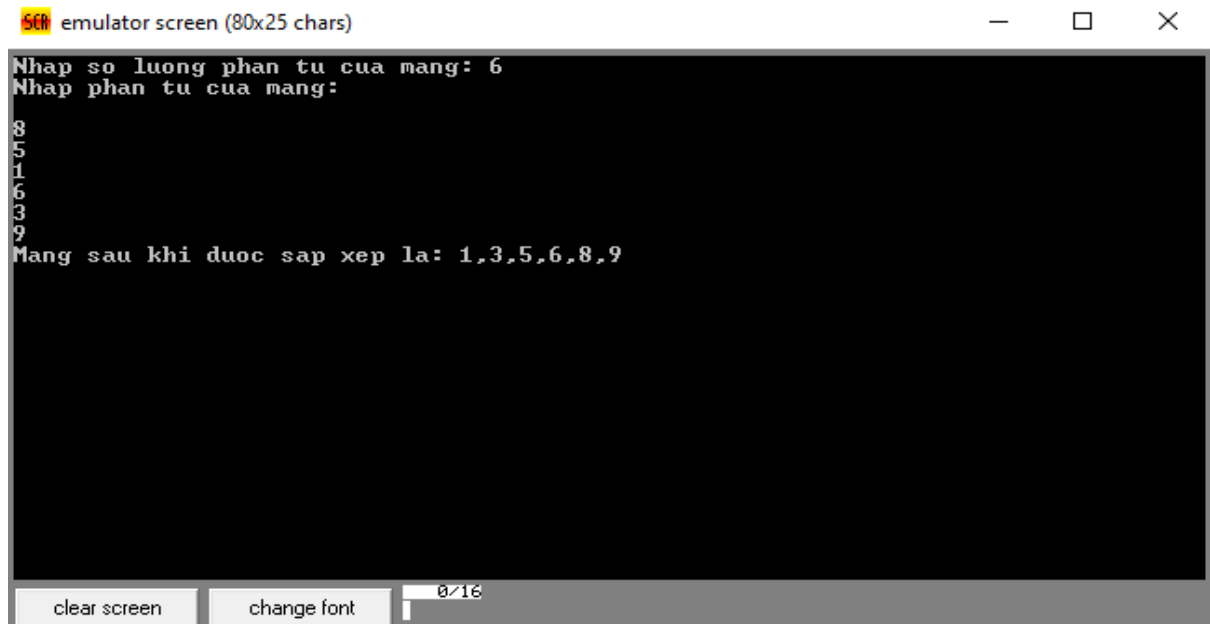
<pre> tb4 db ", \$" under_line DB 0AH,0DH,"\$"  SIZE DW ? temp DW ?  .CODE MAIN PROC      MOV AX,@DATA     MOV DS,AX     MOV AL , ARRAY OUTPUT_TB1:      MOV AH,9     LEA DX,tb1     INT 21h      MOV AH,01H     CALL INPUT_NUMBER     MOV SIZE,BX  OUTPUT_TB2:     MOV BX ,00H      MOV AH,9     LEA DX,under_line     INT 21h      MOV AH,9     LEA DX,tb2     INT 21h      MOV AH,9     LEA DX,under_line     INT 21h  INPUT_ARRAY:      MOV AH,9     LEA DX,under_line     INT 21h      PUSH CX     CMP BX, CX </pre>	<p>Xuống dòng Kích thước mảng Biến tạm temp</p> <p>Phần .code Hàm chính</p> <p>AX = @DATA DS = AX AL = ARRAY</p> <p>Hàm ngắt in ra màn hình In ra màn hình</p> <p>AH = 1 Hàm nhập từ bàn phím SIZE = BX</p> <p>BX = 0</p> <p>Hàm ngắt in ra màn hình Xuống dòng Hàm in ra màn hình</p> <p>Lưu vào CX So sánh BX = CX Bằng thì nhảy</p>
---	--

<pre> JE OUTPUT_SORT PUSH BX CALL INPUT_NUMBER  STOP_INPUT: POP BX MOV SI, BX MOV ARRAY[SI], CL INC BX POP CX JMP INPUT_ARRAY  OUTPUT_SORT: CALL SORTING  MOV AH,9 LEA DX,tb3 INT 21h MOV AX , 00H  CALL OUTPUT  MAIN ENDP ;/////////////////////////////////  INPUT_NUMBER PROC MOV BX,0h MOV CX, 0h next_digit: MOV AH, 00h INT 16h MOV AH, 0Eh INT 10h CMP AL, 13 JNE next JMP STOPINPUT_NUMBER next: MOV BL,AL cmp BX,'0' JB not_minus CMP BX, '9' JA not_minus </pre>	<p>Gọi hàm nhập (nhập các phần tử)</p> <p>SI = BX  ARRAY[SI]= CL  Tăng BX lên 1  Nhảy đến INPUT_ARRAY để nhập tiếp phần tử tiếp theo</p> <p>Gọi hàm sort</p> <p>Hàm in ra màn hình</p> <p>AX = 0</p> <p>Hàm xuất</p> <p>Kết thúc hàm chính</p> <p>Hàm nhập INPUT  BX = 0  CX = 0</p> <p>AH = 0  Interupt  AH = 0EH  Interupt  So sánh AL = 13 (Enter)  Nhảy nếu không bằng  Nhảy  BL = AL  So sánh BX &gt;= '0'(so sánh ascii)</p> <p>Nhảy nếu không đúng  So sánh BX &lt;= '9'(so sánh ascii)  Nhảy nếu không đúng</p>
--	---

<pre> SUB BX,30h MOV AX, CX MOV CX, 0AH MUL CX MOV CX, AX ADD CX, BX JMP next_digit not_minus: MOV AH,4CH INT 21h RET STOPINPUT_NUMBER: RET INPUT_NUMBER ENDP ;///////////////////////////////// SORTING PROC PUSHA MOV CX,CS:SIZE MOV BX,CX SUB BX,1 MOV SI,1 FOR_LOOP: MOV DL,ARRAY[SI] MOV DI,SI SUB DI,1 WHILE_LOOP: CMP DL,ARRAY[DI] JGE EXIT_WHILE_LOOP MOV AL,ARRAY[DI] MOV ARRAY[DI+1],AL SUB DI,1 CMP DI,0 JGE WHILE_LOOP EXIT_WHILE_LOOP: MOV ARRAY[DI+1],DL ADD SI,1 DEC CX CMP CX,1 JNE FOR_LOOP SORT_DONE: POPA RET 4  SORTING ENDP ;///////////////////////////////// OUTPUT_C PROC </pre>	<pre> BX – 30H(Chuyển về mã ascii) AX = CX CX = 10 AX = AX*10 CX = AX CX = BX (Cộng số còn lại) Nhảy  Interupt  Quay về từ lệnh gọi  Kết thúc hàm  Hàm sort Luu register CX = SIZE BX = SIZE BX = SIZE – 1 SI = 1 DL = ARRAY[SI] DI = SI  DI = DI – 1  So sánh DL &gt;= ARRAY[DI] Đúng thì nhảy AL = ARRAY[DI] ARRAY[DI+1] = AL DI = DI – 1 So sánh DI &gt;= 0 Đúng thì nhảy  ARRAY[DI+1] = DL SI = SI + 1 Giảm CX So sánh CX = 1 Sai thì nhảy  Kết thúc hàm sort </pre>
--	--

<pre> MOV CS:temp, SI POP SI PUSH AX nextchar: MOV AL, CS:[SI] INC SI CMP AL, 0 JZ print MOV AH, 0Eh INT 10h JMP nextchar print: POP AX PUSH SI MOV SI, CS:temp RET ;//////////////////// OUTPUT PROC          CMP AX, CX         JE not_minus          MOV SI, AX         MOV BX, 00H         MOV BL, ARRAY[SI]         PUSH CX         PUSH AX          CMP BX, 0AH         JNB OUTPUT_NEXT         ADD BX, 30h         MOV DX, BX         MOV AH, 02H         INT 21H         JMP OUT_1 OUTPUT_NEXT:         MOV CX, 00H         MOV AX, BX DISPLAY:         MOV BX, 10D         MOV DX, 00H         DIV BX         ADD DX, 30H         PUSH DX         INC CX         CMP AX, 0 </pre>	<p>Hàm in ra display  Gán temp = SI  Lấy giá trị vào SI  Đưa giá trị AX vào stack</p> <p>Gán AL = temp[SI]  Tăng SI (tăng địa chỉ)  So sánh AL với 0  Nhảy nếu ZF = 1</p> <p>Gán AH = 0Eh để ngắt  Nhảy tới nextchar</p> <p>Kết thúc hàm</p> <p>So sánh AX = CX  Đúng thì nhảy</p> <p>SI = AX  BX = 0  BL = ARRAY[SI]</p> <p>So sánh BX &gt;= 10  Không đúng thì nhảy  BX + 30h (chuyển sang ascii)  DX = BX  Interrupt để in kí tự  Nhảy đến OUT_1</p> <p>CX = 0  AX = BX</p> <p>BX = 10  DX = 0  AX = AX / 10  DX + 30h</p> <p>Tăng CX thêm 1  So sánh AX = 0</p>
--	---

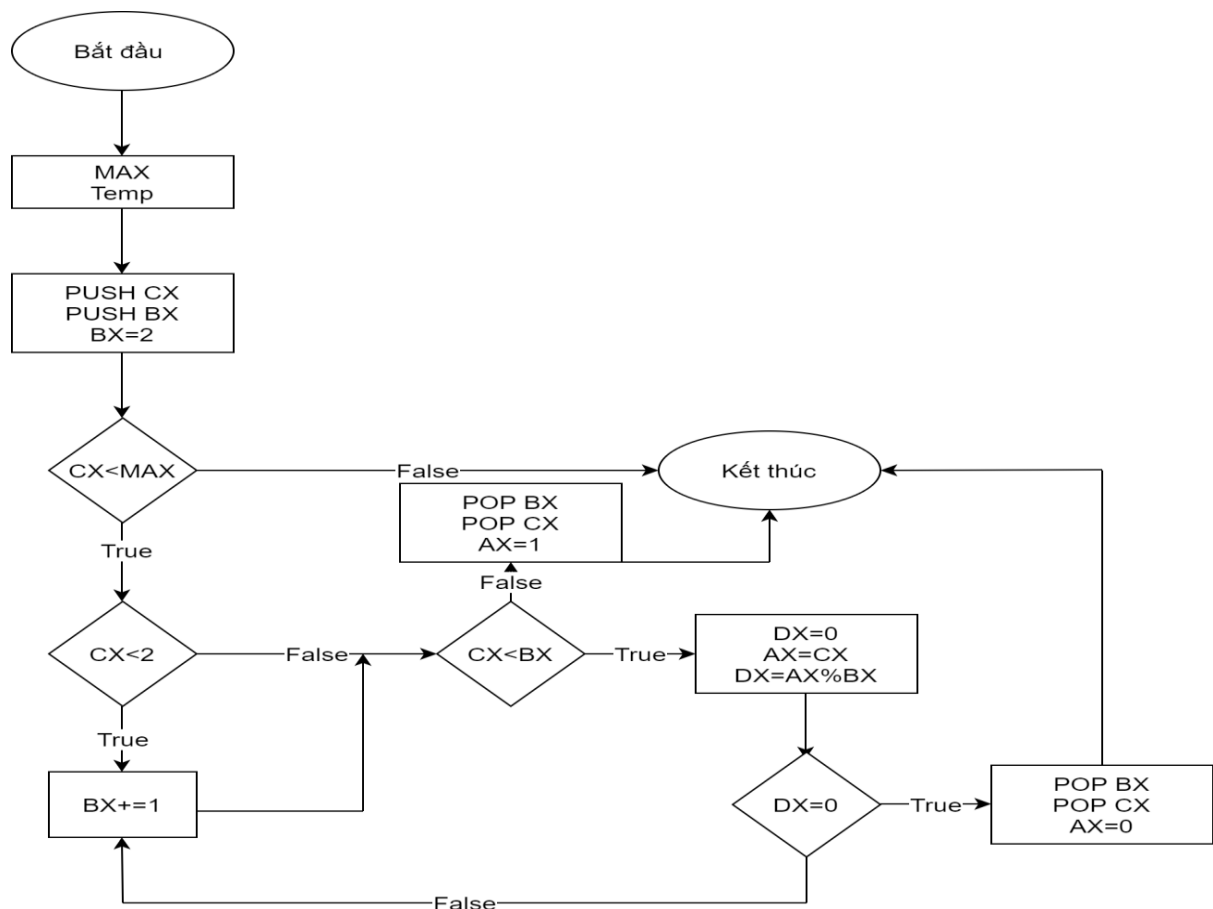




Hình 8 : Mô phỏng kết quả

III. Nhập vào số nguyên N ( $N < 150$ ), in ra các số nguyên tố nhỏ hơn N vào file.txt( mỗi số cách nhau một khoảng trắng)

1. Sơ đồ giải thuật:



Hình 9 : Sơ đồ giải thuật bài 3



Code	Ý nghĩa
<pre> .MODEL SMALL include 'emu8086.inc' org 100h .DATA     tb1 db "Nhap so N: \$"     tb2 db "Cac so nguyen to: \$"     under_line DB 0AH,0DH,"\$"     space db " \$"     outfile DB "c:\result.txt",0     MAX dw ?     temp db ?  .CODE MAIN PROC      MOV AX,@DATA     MOV DS,AX      DISPLAY_TB1:      MOV AH,9     LEA DX,tb1     INT 21h     CALL INPUT_NUMBER      MOV AH,9     LEA DX,under_line     INT 21h     DISPLAY_TB2:     MOV AH,9     LEA DX,tb2     INT 21h      PUSH CX     mov ah, 3ch     mov cx, 0     mov dx, offset outfile     int 21h     jc EXIT     mov MAX, ax     POP CX </pre>	<p>Phần .Data Khai báo nhập xuất</p> <p>Phần xuống dòng Phần khoảng trắng File để ghi kết quả vào Biến MAX Biến tạm Temp</p> <p>Phần Code Hàm chính</p> <p>Hàm thông báo nhập số</p> <p>Dùng hàm ngắt để in ra chuỗi cần thông báo</p> <p>Gọi hàm input_number để nhập số từ bàn phím</p> <p>Dùng hàm ngắt để hiển thị xuống dòng</p> <p>Dùng hàm ngắt để thông báo in các số nguyên tố</p> <p>Lưu CX AH = 3CH CX = 0 DX, offset outfile</p> <p>So sánh AX với 0, đúng thì thoát CT MAX=ax Lấy giá trị CX</p>

MOV BX, 00H	BX=0
LOOP_PRINT: PUSH CX MOV CX, BX	Vòng lặp in ra các số nguyên tố Lưu giá trị của CX CX=BX
CALL PRIME_FUNCTION POP CX	Hàm xử lý nguyên tố Lấy giá trị của CX
CMP AX, 01H	So sánh AX với 01
JE PRINT JMP NEXT	Nếu AX=1 thì nhảy tới hàm print Nếu không thì nhảy tới hàm NEXT
MAIN ENDP ;////////////////////////////////////	Kết thúc hàm chính
INPUT_NUMBER PROC	Hàm nhập giá trị
MOV CX, 0H MOV BX, 0H	CX=0 BX=0
READ: MOV AH,01H INT 21H	AH=01 Interrupt
CMP AL,0DH	So sánh AL với 0Dh
JZ EXIT1	Nhảy nếu =0
MOV BL,AL	BL = AL
cmp BX,'0' JB EXIT cmp BX,'9' JA EXIT	So sánh BX >= '0'(so sánh ascii) Nhảy nếu không đúng So sánh BX <='9'(so sánh ascii) Nhảy nếu không đúng
SUB BX, 30H	BX – 30H(Chuyển về mã ascii)
MOV AX, CX MOV CX, 0AH	AX = CX CX = 10
MUL CX MOV CX, AX ADD CX, BX	AX = AX*10 CX = AX CX = BX (Cộng số còn lại)
JMP READ EXIT1: RET	Nhảy



<pre> DISPLAY_NUMBER PROC     CMP BX, 0AH     JB LESS_THAN_10     MOV CX, 00H     MOV AX, BX     CONTINUE_DISPLAY:     MOV BX, 10D     MOV DX, 00H     DIV BX     ADD DX, 30H     PUSH DX     INC CX     CMP AX, 0     JE DISPLAY_END     JMP CONTINUE_DISPLAY  LESS_THAN_10:     ADD BX, 30h      CALL WRITE_FILE      MOV DX, BX     MOV AH, 02H     INT 21H     RET  DISPLAY_END:     POP DX     MOV BX, DX      CALL WRITE_FILE      MOV AH, 02H     INT 21H      LOOP DISPLAY_END      RET DISPLAY_NUMBER ENDP ;///////////////////////////////// </pre>	<p>So sánh BX với 10 Nhảy nếu BX&lt;10 CX=0 AX=BX</p> <p>BX=10 DX=0 AX = AX/BX DX + 30 (DX = AX%BX) Lưu DX CX++ So sánh AX với 0 Nhảy nếu bằng Nhảy CONTINUE_DISPLAY</p> <p>Hàm nhỏ hơn 10 BX+=30h</p> <p>Gọi hàm ghi vào file</p> <p>DX=BX Interrupt in kí tự Trở về hàm chính</p> <p>Lấy DX BX=DX</p> <p>Gọi hàm ghi file</p> <p>Interrupt in kí tự</p> <p>Kết thúc hàm</p>
---	---

<p>WRITE_FILE PROC</p> <p>PUSH BX  PUSH cx  PUSH DX  mov ah, 40h  MOV temp, BL  mov bx, MAX  mov dx, offset temp  mov cx, 01h  int 21h  POP DX  POP cx  POP BX</p> <p>RET</p> <p>WRITE_FILE ENDP  ;/////////////////////////////////  PRINT:</p> <p>PUSH CX  PUSH BX  CALL DISPLAY_NUMBER</p> <p>mov ah, 40h  MOV temp, 32d  mov bx, MAX  mov dx, offset temp  mov cx, 01h  int 21h  POP BX  POP CX</p> <p>MOV AH,9  LEA DX,space  INT 21h</p> <p>NEXT:  INC BX</p> <p>LOOP LOOP_PRINT</p>	<p>Hàm ghi vào file</p> <p>Lưu giá trị của BX  Lưu giá trị của CX  Lưu giá trị của DX  Ah=40h  Temp=BL  Bx =MAX  Thực hiện ghi  Interrupt</p> <p>Lấy giá trị của BX  Lấy giá trị của CX  Lấy giá trị của DX</p> <p>Trở về hàm chính</p> <p>Kết thúc hàm ghi</p> <p>Lưu giá trị CX  Lưu giá trị BX  Gọi hàm DISPLAY_NUMBER</p> <p>Ah=40h  Temp=32  Bx=MAX  Thực hiện ghi  Interrupt</p> <p>Lấy giá trị của BX  Lấy giá trị của CX</p> <p>In ra kí tự khoảng trắng</p> <p>BX++</p>
--	--

