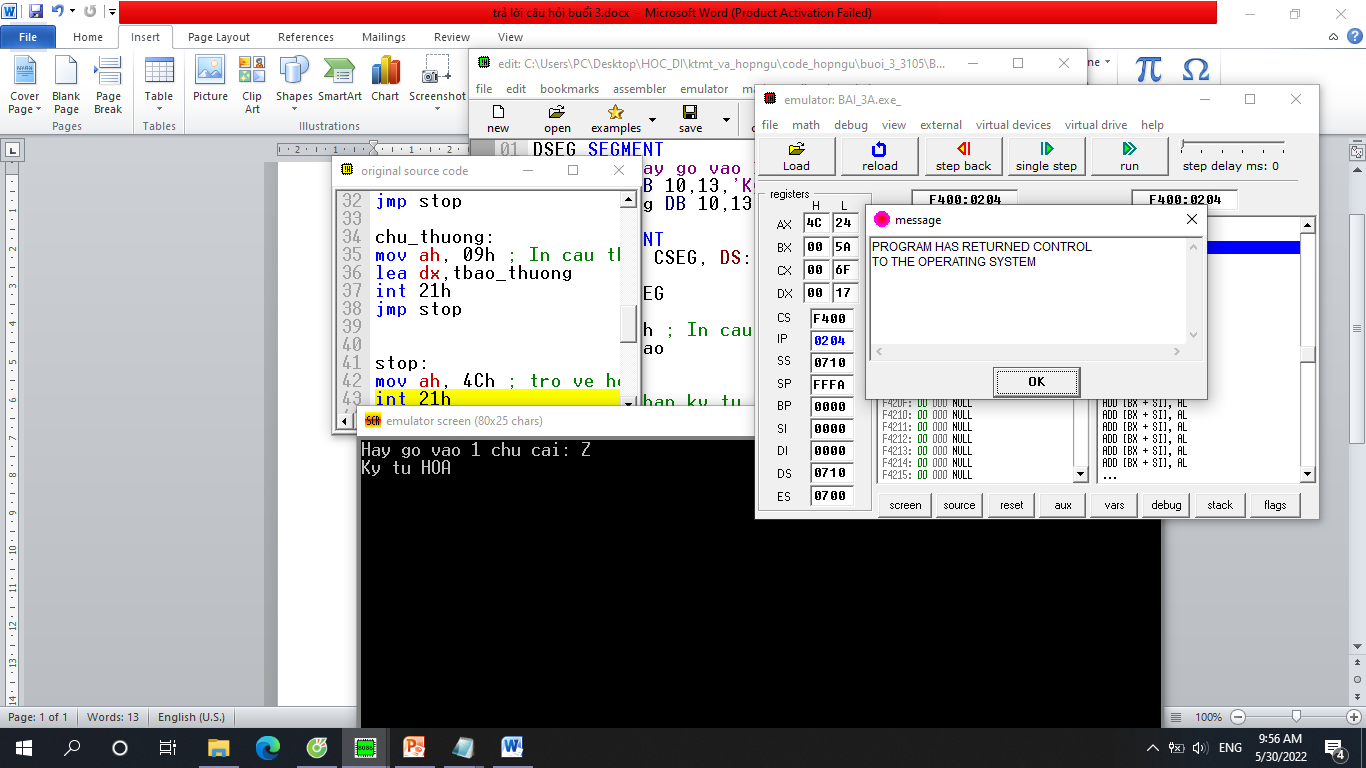
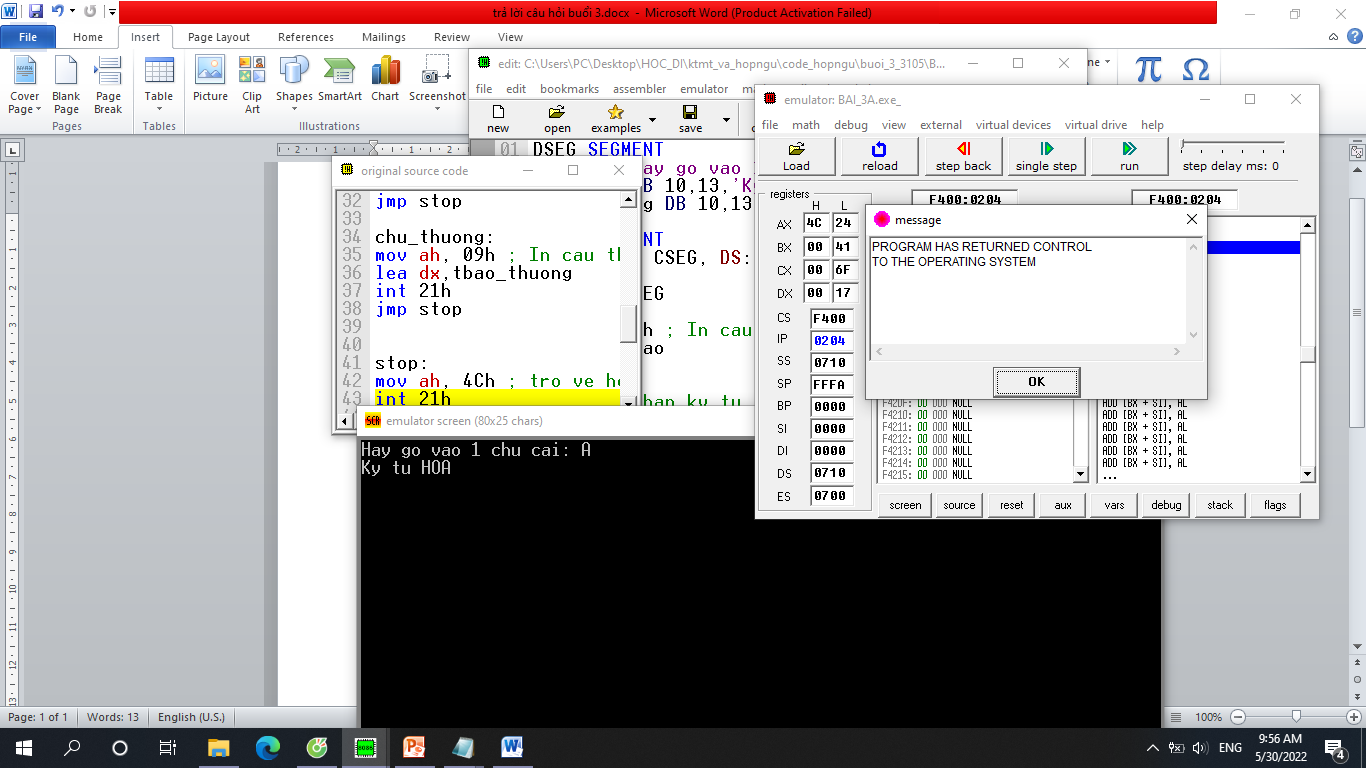
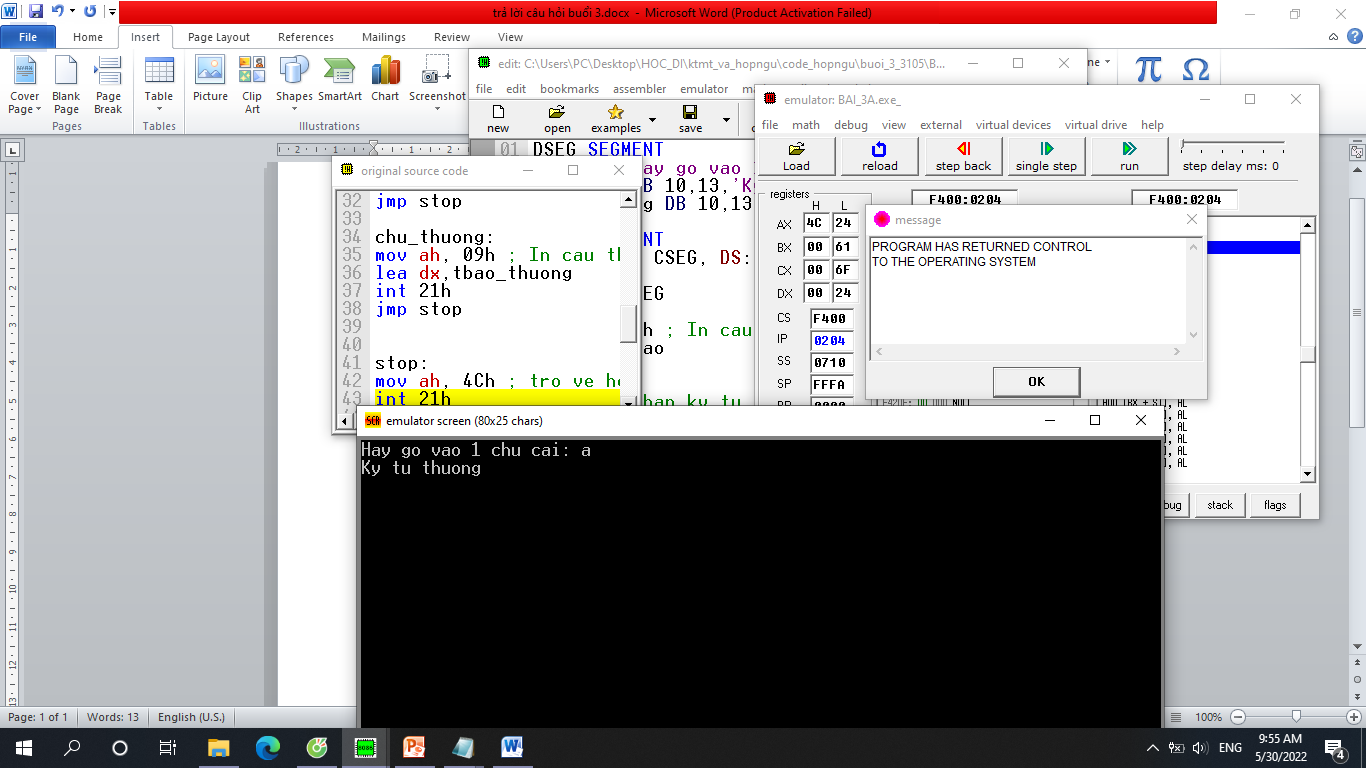
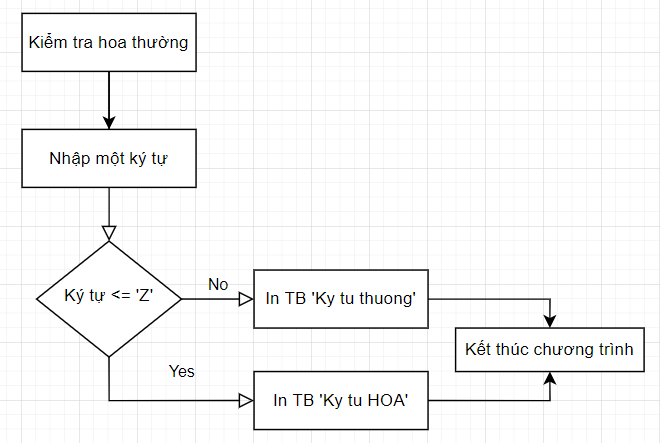
Bài 3a:

Các kết quả cho từng trường hợp:

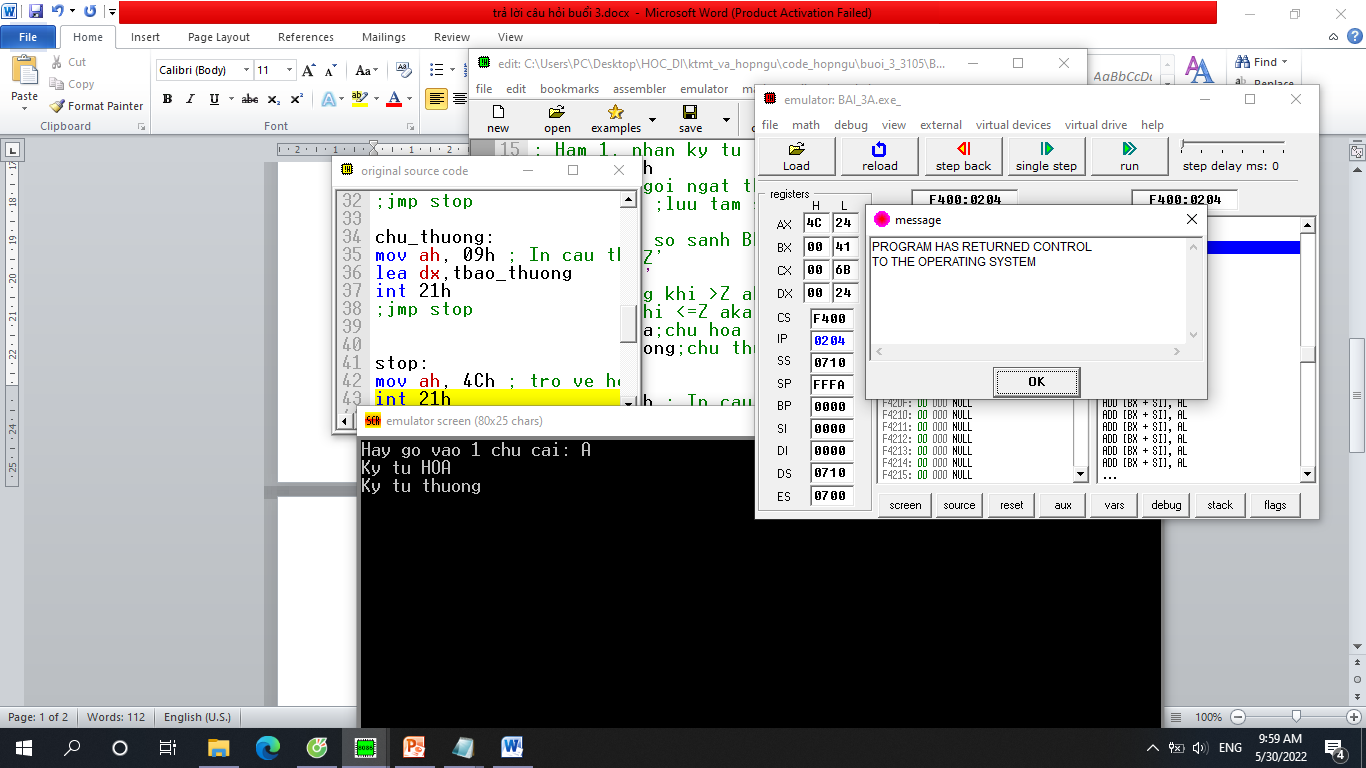


Lưu đồ điều khiển



1. Tại sao cần phải có lệnh JMP EXIT? Nếu không có lệnh ấy thì chương trình thực hiện như thế nào? Chạy chương trình để kiểm chứng.

Nếu không có lệnh JMP EXIT thì chương trình sẽ tự động chạy từ nhãn này sang nhãn khác, gây ra sai lệch về luồng điều khiển mình muốn



1. Thay lệnh JA NHAN bằng lệnh JNA NHAN. Sửa chương trình sao cho kết quả không thay đổi.

JA là “nhảy nếu lớn hơn” còn JNA là “nhảy nếu không lớn hơn” (nhỏ hơn hoặc bằng). JNA tương tự với JBE, nên có thể gán JNA cho nhãn “chu\_hoa” còn JA cho nhãn “chu\_thuong”

1. Khi ký tự nhập vào không phải là chữ cái thì kết quả in ra màn hình là gì? Tại sao?

Vẫn in ra ‘Ky tu HOA’ hoặc ‘Ky tu thuong’ vì hiện tại bài chỉ so sánh với mã ASCII của ký tự ‘Z’ chứ chưa có xử lý ngoại lệ bằng việc so sánh với ký tự ‘A’ hay ‘z’

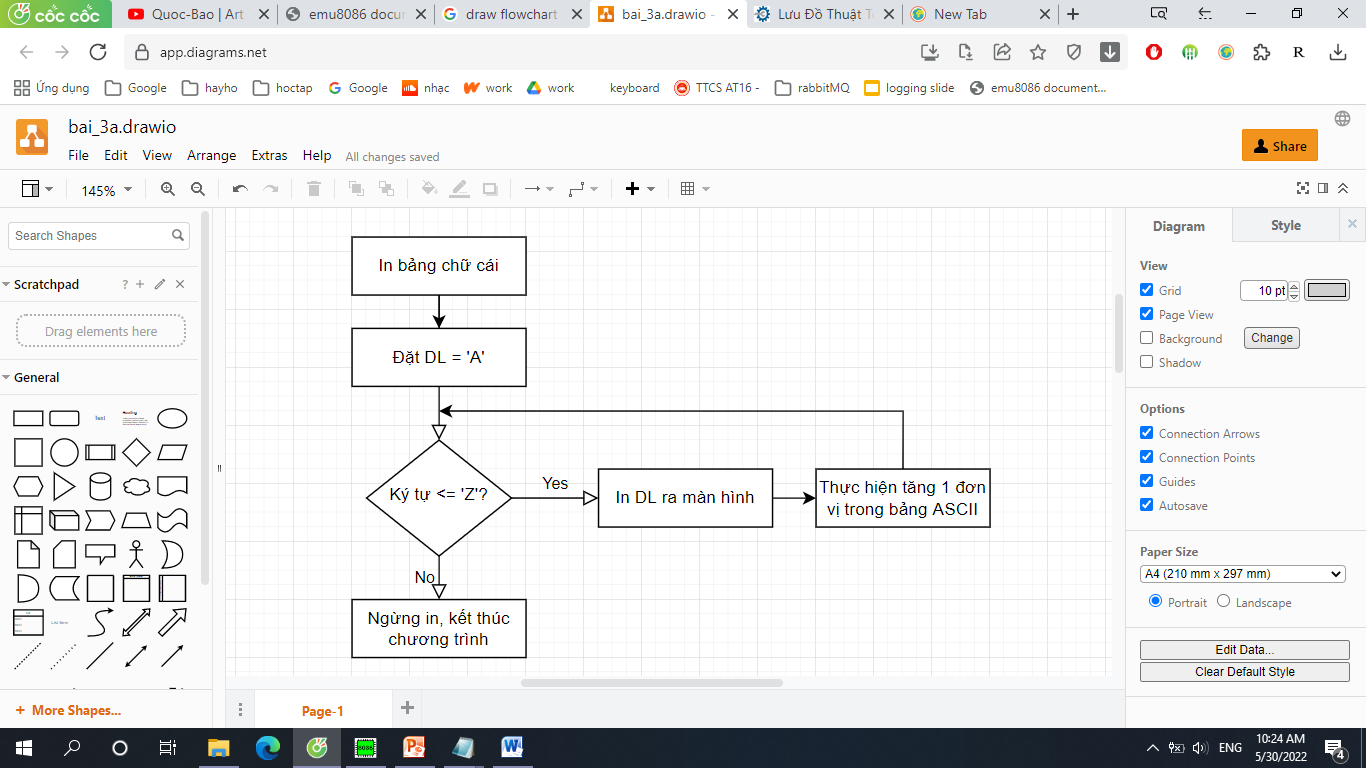
Bài 3b:

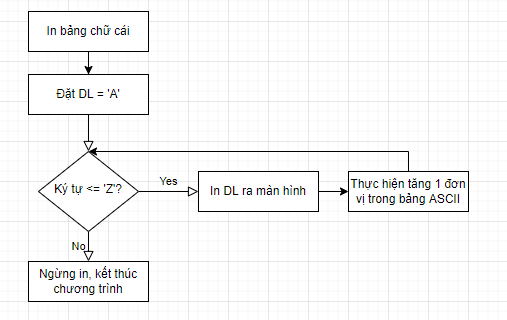
|  |  |  |
| --- | --- | --- |
| Chương trình gốc | In liền mạch từ Z đến A | In có khoảng trắng (Z 🡪 A) |
| CSEG SEGMENT  ASSUME CS: CSEG  start:  mov dl, 'A' ; Dl chua ky tu dau tien 'A'  nhan:  mov ah, 02h ; in ky tu trong DL ra man hinh  int 21h  inc dl ; DL chua ky tu ke can in; DL = DL+1  cmp dl, 'Z' ; So sanh voi ky tu 'Z'  jna nhan ; Neu <= 'Z' thi tiep tuc in  mov ah, 08h ; Neu > 'Z' thi thoat (ngung in)  int 21h  mov ah, 4Ch  int 21h  CSEG ENDS  END start | CSEG SEGMENT  ASSUME CS: CSEG  start:  mov dl, 'Z' ; Dl chua ky tu dau tien 'Z'  nhan:  mov ah, 02h ; in ky tu trong DL ra man hinh  int 21h  dec dl ; DL chua ky tu ke can in; DL = DL-1  cmp dl, 'A' ; So sanh voi ky tu 'A'  jnb nhan ; Neu <= 'A' thi tiep tuc in  mov ah, 08h ; Neu > 'A' thi thoat (ngung in)  int 21h  mov ah, 4Ch  int 21h  CSEG ENDS  END start | CSEG SEGMENT  ASSUME CS: CSEG  start:  mov dl, 'Z' ; Dl chua ky tu dau tien 'Z'  **nhan:**  **mov ah, 02h ; in ky tu trong DL ra man hinh**  **int 21h**  **mov bl,dl ;tam luu DL vao BL**  **mov dl,20h ;khoang trang**  **int 21h**  **mov dl,bl ;tra lai DL nhu cu**  **dec dl ; DL chua ky tu ke can in; DL = DL-1**  **cmp dl, 'A' ; So sanh voi ky tu 'A'**  **jnb nhan ; Neu <= 'A' thi tiep tuc in**  mov ah, 08h ; Neu > 'A' thi thoat (ngung in)  int 21h  mov ah, 4Ch  int 21h  CSEG ENDS  END start |

Chuyển lệnh lặp bằng lable thành loop:

|  |  |  |
| --- | --- | --- |
| CSEG SEGMENT  ASSUME CS: CSEG  start:  mov dl, 'Z' ; Dl chua ky tu dau tien 'Z'  mov cx, 26  L1:  mov ah, 02h ; in ky tu trong DL ra man hinh  int 21h  mov bl,dl ;tam luu DL vao BL | mov dl,20h ;khoang trang  int 21h  mov dl,bl ;tra lai DL nhu cu  dec dl ; DL chua ky tu ke can in; DL = DL-1  cmp dl, 'A' ; So sanh voi ky tu 'A'  ;jnb nhan ; Neu <= 'A' thi tiep tuc in  loop L1 | mov ah, 08h ; Neu > 'A' thi thoat (ngung in)  int 21h  mov ah, 4Ch  int 21h  CSEG ENDS  END start |

Vòng lặp trong chương trình là đoạn in đậm cột 3; Thuộc kiểu while… do. Lưu đồ chương trình:





Bài 1: Full code:

|  |  |  |  |
| --- | --- | --- | --- |
| DSEG SEGMENT  tbao DB 'Hay go vao 1 chu cai: $'  tbao\_sang DB 10,13, 'Good morning! $'  tbao\_chieu DB 10,13, 'Good afternoon! $'  tbao\_toi DB 10,13, 'Good evening! $'  DSEG ENDS  CSEG SEGMENT  ASSUME CS: CSEG, DS: DSEG | start:  mov ax, DSEG  mov ds, ax  mov ah, 09h ; In cau thong bao ra man hinh  lea dx, tbao  int 21h  ; Ham 1, nhan ky tu tu ban phim  mov ah, 01h  int 21h ; goi ngat thuc hien ham  mov bl, al ;luu tam sang BL | ;thuc hien so sanh  CMP BL,'S'  JE sang  CMP BL,'s'  JE sang  CMP BL,'T'  JE chieu  CMP BL,'t'  JE chieu  CMP BL,'C'  JE toi  CMP BL,'c'  JE toi  sang:  mov ah, 09h ; In cau thong bao ra man hinh  lea dx,tbao\_sang  int 21h  jmp stop | chieu:  mov ah, 09h ; In cau thong bao ra man hinh  lea dx, tbao\_chieu  int 21h  jmp stop  toi:  mov ah, 09h ; In cau thong bao ra man hinh  lea dx, tbao\_toi  int 21h  jmp stop  stop:  mov ah, 4Ch ; tro ve he dieu hanh  int 21h  CSEG ENDS  END start |

Bài 2: Khai báo biến: tbao\_khac DB 10,13, 'Ky tu khac $'

Thêm xử lý ngoại lệ ngay trước khi so sánh với ‘Z’:

|  |  |  |
| --- | --- | --- |
| Nhận diện ngoại lệ: | Ngoại lệ ASCII từ 91-96 | In thông báo ngoại lệ: |
| ;xu ly ngoai le <A & >a  CMP BL,'A'  JB TH\_khac  CMP BL, 'a'  JA TH\_khac | chu\_thuong:  CMP BL,'a'  JB TH\_khac  mov ah, 09h  lea dx, tbao\_thuong  int 21h  jmp stop | TH\_khac:  mov ah, 09h ; In cau thong bao ra man hinh  lea dx,tbao\_khac  int 21h  jmp stop |

Bài 3: Full code:

|  |  |  |
| --- | --- | --- |
| DSEG SEGMENT  tbao DB 'Hay go vao 1 chu cai viet thuong: $'  DSEG ENDS  CSEG SEGMENT  ASSUME CS: CSEG, DS: DSEG  start:  mov ax, DSEG  mov ds, ax  mov ah, 09h ; In cau thong bao ra man hinh  lea dx, tbao  int 21h | ; Ham 1, nhan ky tu tu ban phim  mov ah, 08h  int 21h ; goi ngat thuc hien ham  mov dl, al ;luu sang DL  nhan:  mov ah, 02h ; in ky tu trong DL ra man hinh  int 21h  ;tao khoang trang  mov bl,dl ;tam luu DL vao BL  mov dl,20h ;khoang trang  int 21h  mov dl,bl ;tra lai DL nhu cu | inc dl ; DL chua ky tu ke can in; DL = DL+1  cmp dl, 'z' ; So sanh voi ky tu 'Z'  jna nhan ; Neu <= 'z' thi tiep tuc in  mov ah, 08h ; Neu > 'z' thi thoat (ngung in)  int 21h  mov ah, 4Ch  int 21h  CSEG ENDS  END start |

Bài 4: Full code: Tạo mảng siêu lớn, nhập từng phần tử mảng, đếm số phần tử 🡪 Ra được độ dài chuỗi

|  |  |
| --- | --- |
| DSEG SEGMENT  chuoi DB 100 DUP(0)  length DB 10,13,'Do dai chuoi: $'  DSEG ENDS  CSEG SEGMENT  ASSUME CS: CSEG, DS: DSEG  start:  mov ax, DSEG  mov ds, ax  MOV SI, 0 ; chi so mang  MOV CX, 100 ; do dai mang max  LAP: MOV AH, 1 ; nhap ky tu  INT 21H  CMP AL,0Dh ;check dau enter  JE STOP  MOV chuoi[SI], AL  INC SI  JMP LAP | STOP:  mov ah, 09h ; In cau thong bao ra man hinh  lea dx, length  int 21h  ;in KQ so dem  mov ah, 02h;  mov DX, SI ;hien tai dang bi in thanh ASCII  add DX,30h  int 21h  mov ah,08h  int 21h  mov ah, 4Ch ; tro ve he dieu hanh  int 21h  CSEG ENDS  END start |

Bài 5: Phân biệt chữ hoa – thường – dấu cách – dấu enter; tùy loại mà có cách xử lý để chia làm chuỗi chữ thường và chuỗi chữ hoa;

|  |  |  |  |
| --- | --- | --- | --- |
| DSEG SEGMENT  chuoi DB 100 DUP(0)  inthuong DB 100 DUP(0)  inhoa DB 100 DUP (0)  kq DB 10,13,'Ket qua: $'  DSEG ENDS  CSEG SEGMENT  ASSUME CS: CSEG, DS: DSEG  start:  mov ax, DSEG  mov ds, ax  MOV SI, 0 ; chi so mang  MOV CX, 100 ; do dai mang max  LAP: MOV AH, 1 ; nhap ky tu  INT 21H  CMP AL,0Dh ;check dau enter  JE STOP  CMP AL,20h ;check dau space  JE dau\_trang  CMP AL, 'Z'  JBE chu\_hoa;chu hoa  JA chu\_thuong;chu thuong  next:  MOV chuoi[SI], AL  INC SI  JMP LAP  dau\_trang:  ;nem vao string hoa  mov inhoa[SI],AL  ;nem vao string thuong  mov inthuong[SI],AL  jmp next  chu\_hoa:  ;nem vao string hoa  mov inhoa[SI],AL  ;day la chu hoa, convert thanh chu thuong  ADD AL, 20h  ;nem vao string thuong  mov inthuong[SI],AL  jmp next |  | chu\_thuong:  ;nem vao string thuong  mov inthuong[SI],AL  ;day la chu thuong, convert thanh chu hoa  SUB AL, 20h  ;nem vao string hoa  mov inhoa[SI],AL  jmp next  STOP: ;co duoc chuoi  mov ah, 09h ; In cau thong bao ra man hinh  lea dx, kq  int 21h  mov BX,SI ;cat tam  ;in chuoi thuong  mov CX,SI  INC CX  mov SI,0  mov ah,02h  L1:  mov DL,inthuong[SI]  int 21h  inc SI  loop L1  ;in chuoi hoa  mov SI,BX  mov CX,SI  INC CX  mov SI,0  mov ah,02h  L2:  mov DL,inhoa[SI]  int 21h  inc SI  loop L2    mov ah,08h  int 21h  mov ah, 4Ch ; tro ve he dieu hanh  int 21h  CSEG ENDS  END start |  |