|  |  |
| --- | --- |
| **Họ và tên** |  |
| **MSSV** |  |
| **Lớp** | L08 |

**NỘP BÀI CODE THI CUỐI KỲ HTMT&NNLT**

**Bài 1** (1 điểm): Viết chương trình hợp ngữ LC-3 đếm số lần xuất hiện của một ký tự trong một chuỗi cho trước có nhãn là **MY\_STR**, ký tự cần tìm được nhập từ bàn phím, in kết quả ra màn hình (giả sử rằng **số lần xuất hiện của ký tự ít hơn 100 lần**). Chương trình bắt đầu từ ô nhớ có địa chỉ x3000:

**Bài code:**

|  |  |
| --- | --- |
| .ORIG X3000  IN  ADD R1, R0, #0  LD R2, N  AND R3, R3, #0  LEA R4, MY\_STR  LAP LDR R5, R4, #0  NOT R5, R5  ADD R5, R5, #1  ADD R6, R1, R5  BRz TANG  ADD R4, R4, #1  ADD R2, R2, #-1  BRp LAP  BRnzp KETTHUC  TANG ADD R3, R3, #1  ADD R4, R4, #1  ADD R2, R2, #-1  BRp LAP  KETTHUC LD R2, NASCII  LEA R0, TB1  PUTS  ADD R0, R3, R2  OUT  HALT  N .FILL #100  NASCII .FILL X30  TB1 .STRINGZ "\nSO LAN XUAT HIEN LA: "  **MY\_STR .STRINGZ "AAaBbbbcdefgH"**  .END | - Yêu cầu nhập ký tự cần kiểm tra và lưu vào **R1**  - Gán giá trị N = 100 cho **R2** để đếm ngược vòng lặp “số lần xuất hiện của ký tự ít hơn 100 lần”  - Đưa R3 về giá trị 0 để làm biến đếm  - Lưu vị trí bắt đầu của MY\_STR vào **R4**  - Load ký tự lưu trong ô nhớ có địa chỉ trong **R4** lưu vào **R5**  - Lấy bù 2 của **R5** để tiến hành so sánh bằng phép trừ  - Nếu **R1 – R5 = 0**  thì 2 ký tự trùng nhau thì thực hiện tăng biến đếm **R3**  - Nếu 2 ký tự khác nhau (**R1 – R5 khác 0**) thì **R4 = R4 + 1** để chuyển đến ô nhớ kế tiếp đồng thời **R2 = R2 – 1** đến khi nào về 0 thì kết thúc vòng lặp  - Tăng biến đếm **R3** lên 1 đơn vị  - Chuyển **R4** tới vị trí ô nhớ tiếp theo để xét đồng thời giảm **R2** tương tự như phần trên  - Kết thúc vòng lặp và thông báo kết quả  - Chuyển giá trị trong biến đếm **R3** sang mã ASCII để in ra màn hình  - Đề ghi **MY\_STR** là chuỗi cho trước nên trong bài làm em không có dòng này, để chương trình chạy được thì cần phải khai báo chuỗi MY\_STR |

**Text, letter

Description automatically generatedKết quả:**

* Chuỗi MY\_STR: **AAaBbbbcdefgH**

**Bài 2:** Viết chương trình bằng ngôn ngữ C thực hiện các yêu cầu sau:

1. Viết hàm int kt\_songuyento (int p) thực hiện kiểm tra xem số p có phải là số nguyên tố hay không. Nếu p là số nguyên tố thì trả về 1, ngược lại trả về 0. (1 điểm)
2. Viết hàm int kt\_xuathien (int q, int a[], int k) thực hiện kiểm tra xem số q đã xuất hiện trong mảng a[0] đến a[k-1] hay chưa. Nếu q đã xuất hiện thỉ trả về 0, nếu chưa xuất hiện thì trả về 1. (1 điểm)
3. Viết chương trình cho phép người dùng thực hiện các yêu cầu sau theo thứ tự ( có sử dụng các hàm phía trên) (1 điểm)

* Nhập số nguyên N trong khoảng từ 2 đến 1000, nếu người dùng nhập sai thì bắt nhập lại cho đến khi thoả mãn.
* Nhập một mảng gồm N sô nguyên.
* Kiểm tra xem có bao nhiêu số nguyên tố khác nhau trong mảng đó. In ra kết quả đếm và các số nguyên tố trong mảng (không lặp lại)

**Bài Code:**

|  |  |
| --- | --- |
| #include<stdio.h>  #include<conio.h>  #include<math.h>  int kt\_songuyento(int p){  if (p<2) return 0;  if (p==2) return 1;  int dem = 0;  int e = sqrt(p);  for(int i=2; i<= e; i++){  if(p%i==0) dem++;  }  if (dem == 0) return 1;  else return 0;  }  int kt\_xuathien(int q, int a[], int k){  for(int i =0; i<k; i++){  if (q == a[i]) return 0;  }  return 1;  }  int main(){  int a[1000];  int N;  do {  printf("\nMoi nhap so nguyen N: ");  scanf("%d",&N);  }while ((N<2)||(N>1000));  int b[1000], dem = 0;  for (int i = 0; i< N; i++){  printf("\na[%d]= ",i);  scanf("%d",&a[i]);  if((kt\_songuyento(a[i]))&&(kt\_xuathien(a[i],a,i)))  {b[dem] = a[i]; dem++;}  }  printf("\nMang co %d so nguyen to khac nhau la: \n",dem);  for (int i = 0; i< dem; i++ ) printf("%d\t", b[i]);  return 0;  } | Kiểm tra số nguyên tố:  - Nếu p < 2 lập tức trả về 0(không phải)  - Nếu p = 2 lập tức trả về 1 (là số nguyên tố)  - Còn lại p>2 chỉ cần kiểm tra trong khoảng từ 2 đến căn bậc 2 của p xem p có chia hết cho số nào không, nếu không thì là số nguyên tố. Không bắt buộc phải kiểm tra hết đến p-1 (tính đối xứng qua căn bậc 2) |
| Hàm kiểm tra xem q đã xuất hiện trong mảng từ a[0] đến a[k-1] hay chưa |
| Do phân bổ thời gian không hợp lý nên em làm chưa kịp xong ý này (phần bôi xanh là phần em còn thiếu) mong thầy có thể nhẹ tay 😥 |

**Text

Description automatically generatedKết quả:**