Các hàm hỗ trợ thao tác trên chuỗi ký tự

1.1. Thư viện <stdio.h>

1.1.1. Hàm `scanf` (nhập chuỗi)

- Hàm thực hiện nhận chuỗi ký tự được nhập từ bàn phím.
- Lưu ý: do hàm `scanf` khi gặp phím `space`, `tab`, `new line`, `Enter` thì dừng, cho nên chỉ dùng hàm `scanf` để nhập chuỗi không có khoảng trắng.
- Ví dụ: `scanf ("%s" , & Hoten);`

1.1.2. Hàm `printf` (Xuất Chuỗi)

- Hàm thực hiện xuất chuỗi ra màn hình.
- Ví dụ:

```
#include <stdio.h>
#include <string.h>
void main (void)
{
     char name[20];
     printf ("Enter a name: "); // printf (xuất chuỗi, không xuống dòng)
     scanf ("%s", name); // không sử dụng & trước tên biến name
     printf ("Hello %s\n", name); // printf (xuất chuỗi kèm xuống dòng)
}
```

1.2. Thư viện <string.h>

1.2.1. Hàm clrscr ()

- Dùng để xoá màn hình.
- *Cú pháp*: `void clrscr (void)`

1.2.2. Hàm `gets`

- Hàm thực hiện nhận chuỗi ký tự được nhập từ bàn phím.
- Ví dụ: `gets (Hoten);`
- Tiếp nhận được các phím `space`, `tab`, `new line`; gặp `Enter` thì dừng.
- Phải khai báo hàm xóa bộ đệm bàn phím trước khi dùng hàm `gets`: `fflush (stdin)` hay `flushall()`.

1.2.3. Hàm `puts`

- Hàm thực hiện xuất chuỗi xong tự động xuống dòng.
- Ví dụ: `puts (Hoten);`

1.2.4. Hàm kbhit

- Kiểm tra bộ đệm bàn phím.
- *Cú pháp*: `int kbhit (void)`
- Hàm trả về giá trị khác không nếu bộ đệm bàn phím khác rỗng, trả về giá trị không nếu ngược lại.

1.2.5. Hàm gotoxy()

- Dùng để di chuyển con trỏ (màn hình) đến vị trí (x,y).
- Trong đó x là số hiệu cột có giá trị từ 1 đến 80, và y là số hiệu dòng có giá trị từ 1 đến 25.
- Cú pháp: `void gotoxy(int x, int y)`

1.2.6. Hàm strcat

- Dùng để nối hai chuỗi lại với nhau.
- *Cú pháp*: `char * strcat(char* s1, char* s2)`
- Hàm có công dụng ghép nối hai chuỗi `s1` và `s2` lại với nhau; kết quả ghép nối được chứa trong `s1`.
- *Ví dụ*:

```
#include "stdio.h"
#include "string.h"
void main()
{
     char *s1 = “Khoa ”;
     char *s2 = "CNTT";
     strcat(s1, s2);
     printf("%s",s1); // Kết quả: Khoa CNTT
}
```

1.2.7. Hàm strncat

- Dùng để nối `n` ký tự đầu tiên của chuỗi `s2` vào chuỗi `s1`.
- *Cú pháp*: `char * strncat(char* s1, char* s2, int n)`
- *Ví dụ*:

```
#include "stdio.h"
#include "string.h"
void main()
{
     char *s1 = “Khoa ”;
     char *s2 = "CNTT";
     strncat(s1, s2, 2);
     printf("%s",s1);     // Kết quả: Khoa CN
}
```

1.2.8. Hàm strchr

- Tìm lần xuất hiện đầu tiên của ký tự c trong chuỗi s.
- *Cú pháp*: `char* strchr (char* s, char c)`
- Nếu tìm thấy hàm trả về vị trí tìm thấy của ký tự `c` trong chuỗi `s`, trái lại hàm trả về giá trị `NULL`.
- *Ví dụ*:

```
#include "stdio.h"
#include "string.h"
void main()
{
     char s[15];
     char *ptr, c = 'm';
     strcpy(s, "Vi du tim ky tu");
```

```
        ptr = strchr(s, c);
        if (ptr)
          printf("Ky tu %c xuat hien tai vi tri %d",c, ptr-s);
        else
           printf("Khong tim thay");
        //Kết quả: Ky tu m xuat hien tai vi tri 8
    }
```

1.2.9. Hàm strcmp

- *Cú pháp*: `int strcmp (char* s1, char* s2)`
- Hàm có công dụng so sánh hai chuỗi `s1` và `s2`.
  - Nếu hàm trả về giá trị `<0` khi chuỗi `s1` nhỏ hơn chuỗi `s2`.
  - Nếu hàm trả về giá trị `=0` khi chuỗi `s1` bằng chuỗi `s2`.
  - Nếu hàm trả về giá trị `>0` khi chuỗi `s1` lớn hơn chuỗi `s2`.
- *Ví dụ*:

```
#include "stdio.h"
#include "string.h"
void main()
{    char *s1 = “abcd”;
     char *s2 = "abCD";
     if(strcmp(s1, s2)==0)
         printf(Giong nhau");
     else
        printf( “Khac nhau”; // Kết quả: Khac nhau
}
```

1.2.10. Hàm stricmp

- Hàm này thực hiện việc so sánh trong `n` ký tự đầu tiên của 2 chuỗi `s1` và `s2`. Không phân biệt giữa chữ thường và chữ hoa.
- Kết quả trả về tương tự như kết quả trả về của hàm `strcmp()`
- *Cú pháp*: `int stricmp (const char *s1, const char *s2)`
- *Ví dụ*:

```
#include "stdio.h"
#include "string.h"
void main()
{
     char *s1 = “aBcd”;
     char *s2 = "Abef";
     if(strnicmp(s1, s2, 2)==0)
         printf("Giong nhau";
     else
        printf(“Khac nhau”;
     //Kết quả: Giong nhau
}
```

1.2.11. Hàm strncmp

- Tương tự như hàm strcmp(), nhưng chỉ so sánh `n` ký tự đầu tiên của 2 chuỗi `s1` và `s2`.
- *Cú pháp*: `int stricmp (const char *s1, const char *s2)`
- *Ví dụ*:

```
#include "stdio.h"
#include "string.h"
void main()
{
     char *s1 = “abcd”;
     char *s2 = "abef";
     if(strncmp(s1, s2, 2)==0)
         printf(“Giong nhau";
     else
        printf(“Khac nhau”;
     //Kết quả: Giong nhau
}
```

1.2.12. Hàm strcpy

- Hàm được dùng để sao chép toàn bộ nội dung của chuỗi nguồn (`Src`) vào chuỗi đích (`Des`).
- *Cú pháp*: `char *strcpy (char *Des, const char *Src)`
- *Ví dụ*:

```
#include "stdio.h"
#include "string.h"
void main()
{
     char s[50];
     strcpy(s,"Truong Dai hoc Quoc Te Hong Bang");
     printf("\nXuat chuoi : %s",s);
}
```

1.2.13. Hàm strncpy

- Hàm này cho phép chép `n` ký tự đầu tiên của chuỗi nguồn (`Src`) sang chuỗi đích (`Des`).
- *Cú pháp*: `char *strncpy(char *Des, const char *Source, size_t n)`
- *Ví dụ:*

```
char dest[4];
char *src = "abcdefghi";
strncpy(dest, src, 3);
printf("%s",dest);  //Kết quả: abc
```

1.2.14. Hàm strlen

- Lấy chiều dài chuỗi với hàm.
- *Cú pháp*: `int strlen (const char *s);`
- *Ví dụ 1*:

```
#include <stdio.h>
#include <string.h>
void main()
{
     char string [ ] = "Sai gon";
     printf("%d\n", strlen(string)); // kết quả 7
     getch ();
}
```

- *Ví dụ 2*: Sử dụng hàm strlen xác định độ dài một chuỗi nhập từ bàn phím.

```
#include<conio.h>
```

```
#include<stdio.h>
#include<string.h>
int main()
{
     char Chuoi [255];
     int Dodai;
     printf("Nhap chuoi: ");
     gets(Chuoi);
     Dodai = strlen(Chuoi)
     printf("Chuoi vua nhap: ");
     puts(Chuoi);
     printf("Co do dai %d",Dodai);
     getch();
     return 0;
}
```

- *Ví dụ 3*: Gán một chuỗi vào chuỗi khác bằng cách gán từng ký tự trong chuỗi.

```
#include<conio.h>
#include<stdio.h>
#include<string.h>
void main()
{
     clrscr() ;
     char newstr [35];
     char str[] = "Viet Nam";
     for(int i = 0; i<strlen(str); i++)
          newstr[i] = str[i];
     newstr[i] = '\0';
     getch () ;
}
```

1.2.15. Hàm strstr()

- Hàm `strstr()` được sử dụng để tìm kiếm sự xuất hiện đầu tiên của chuỗi `s2` trong chuỗi `s1`.
- *Cú pháp*: `char *strstr(const char *s1, const char *s2)`
- Kết quả trả về của hàm là một con trỏ chỉ đến phần tử đầu tiên của chuỗi `s1` có chứa chuỗi `s2` hoặc giá trị `NULL` nếu chuỗi s2 không có trong chuỗi s1.
- *Ví dụ*: Viết chương trình sử dụng hàm `strstr()` để lấy ra một phần của chuỗi gốc bắt đầu từ chuỗi "dai".

```
#include<stdio.h>
#include<string.h>
int main()
{
     char *s1 = "Borland International";
     char *s2 = "nation", *ptr;
     ptr = strstr(s1, s2);
     printf("Chuoi con: %s", ptr);
     //Kết quả: Chuoi con: national
}
```

1.2.16. Hàm strupr()

- Hàm `strupr()` được dùng để chuyển đổi chuỗi chữ thường thành chuỗi chữ hoa, kết quả trả về của hàm là một con trỏ chỉ đến địa chỉ chuỗi được chuyển đổi.
- *Cú pháp*: char *strupr(char *s)
- *Ví dụ*: Viết chương trình nhập vào một chuỗi ký tự từ bàn phím. Sau đó sử dụng hàm `strupr()` để chuyển đổi chúng thành chuỗi chữ hoa.

```
#include<conio.h>
#include<stdio.h>
#include<string.h>
void main()
{
     char Chuoi[255],*s;
     printf("Nhap chuoi: ");
     gets(Chuoi);
     s=strupr(Chuoi) ;
     printf("Chuoi chu hoa: ");
     puts(s);
     getch();
}
```

1.2.17. Hàm strlwr()

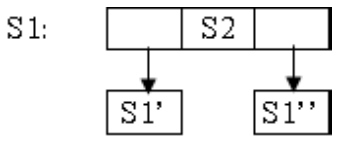
- Muốn chuyển đổi chuỗi chữ hoa thành chuỗi toàn chữ thường, ta sử dụng hàm `strlwr()`, các tham số của hàm tương tự như hàm `strupr()`
- *Cú pháp*: `char *strlwr (char *s)`

1.2.18. Hàm atoi(), atof(), atol()

- Để chuyển đổi chuỗi ra số, với kiểu dữ liệu tùy thuộc hàm được dùng.
- Cần khai báo thư viện `<stdlib.h>` khi sử dụng hàm.
- *Cú pháp*:
  - `int atoi(const char *s)` : chuyển chuỗi thành số nguyên
  - `long atol(const char *s)` : chuyển chuỗi thành số nguyên dài
  - `float atof(const char *s)` : chuyển chuỗi thành số thực
- Nếu chuyển đổi không thành công, kết quả trả về của các hàm là 0.

1.2.19. char* strtok (char *s1 , const char *s2)

- Nếu `s2` có xuất hiện trong `s1`: Tách chuỗi s1 thành hai chuỗi: Chuỗi đầu là những ký tự cho đến khi gặp chuỗi `s2` đầu tiên, chuỗi sau là những ký tự còn lại của `s1` sau khi đã bỏ đi chuỗi `s2` xuất hiện trong `s1`.



- Nếu s2 không xuất hiện trong s1 thì kết quả chuỗi tách vẫn là s1.
- Ví dụ :

```
#include <string.h>
#include<stdio.h>
void main()
{
     char input[16] = "abc,d";
     char *p;
     // Lay chuoi dau
```

```
        p = strtok(input, ",");
        if (p)
            printf("S11: %s-",p);
        /*Lay chuoi con lai, tham so dau la NULL*/
        p = strtok(NULL, ",");
        if (p)
            printf("S12: %s", p);
        // Kết quả: S11: abc - S12: d
    }
```

1.2.20. char* strrev(char *s)

- Đảo ngược chuỗi.

1.3. Thư viện < ctype.h>

1.3.1. Hàm toupper()

- Hàm được dùng để chuyển đổi một ký tự thường thành ký tự hoa.
- *Cú pháp*: char toupper (char c)

1.3.2. Hàm tolower()

- Hàm được dùng để chuyển đổi một ký tự hoa thành ký tự thường.
- *Cú pháp*: char tolower (char c)

1.3.3. Hàm isalpha

- Hàm thực hiện kiểm tra xem char_exp có phải là một chữ cái hay không?
- *Cú pháp*: **`int isalpha (char_exp)`**
- *Kết quả trả về*: <>0 khi *char_exp* là một chữ cái hoa
- *Ví dụ*: `int kq= isupper('a');`

1.3.4. Hàm isupper

- Hàm thực hiện kiểm tra xem char_exp có phải là một chữ cái hoa hay không?
- *Cú pháp*: **`int isupper (char_exp)`**
- *Kết quả trả về*: <>0 khi *char_exp* là một chữ cái
- *Ví dụ*: `int kq= isupper('a');`

1.3.5. Hàm islower

- Hàm thực hiện kiểm tra xem char_exp có phải là một chữ cái thường hay không?
- *Cú pháp*: **`int isupper (char_exp)`**
- *Kết quả trả về*: <>0 khi *char_exp* là một chữ cái thường
- *Ví dụ*: `int kq= islower('a');`

1.3.6. Hàm isdigit

- Hàm thực hiện kiểm tra xem char_exp có phải là một ký số hay không?
- *Cú pháp*: **`int isdigit (char_exp)`**
- *Kết quả trả về*: <>0 khi *char_exp* là một ký số
- *Ví dụ*: `int kq= isdigit ('a');`

1.3.7. Hàm isascii

- Hàm thực hiện kiểm tra xem char_exp có phải là một ký tự có mã ASCII<128 hay không?

- *Cú pháp*: **int isascii(*char_exp*)**
- *Kết quả trả về*: <>0 khi *char_exp* là một ký tự có mã ASCII<128.
- *Ví dụ*: `int kq= isascii('a');`

1.3.8. Hàm isspace

- Hàm thực hiện kiểm tra xem char_exp có phải là một khoảng trắng hay không?
- *Cú pháp*: **int isspace (*char_exp*)**
- *Kết quả trả về*: <>0 khi *char_exp* là ký tự khoảng trắng
- *Ví dụ*: `int kq= isspace ('a');`

1.3.9. Hàm isprint

- Hàm thực hiện kiểm tra xem char_exp có phải là một ký tự có thể in được hay không?
- *Cú pháp*: **int isprint (*char_exp*)**
- *Kết quả trả về*: <>0 khi *char_exp* là một ký tự có thể in được.
- *Ví dụ*: `int kq= isprint ('a');`

1.3.10. Hàm iscntrl

- Hàm thực hiện kiểm tra xem char_exp có phải là một ký tự điều khiển hay không?
- *Cú pháp*: **int iscntrl (*char_exp*)**
- *Kết quả trả về*: <>0 khi *char_exp* là một ký tự có thể in được.
- *Ví dụ*: `int kq= iscntrl ('!');`

1.4. Thư viện <stdlib.h>

1.4.1. Hàm atoi

- Hàm thực hiện chuyển một chuỗi sang số nguyên. Việc chuyển đổi sẽ dừng khi gặp ký tự không phải là ký số
- *Cú pháp*: **int atoi(*str_exp*)**
- *Ví dụ*: `int kq= atoi('123a45');`

1.4.2. Hàm atof

- Hàm thực hiện chuyển một chuỗi sang số double. Việc chuyển đổi sẽ dừng khi gặp ký tự không thể chuyển sang dạng double được
- *Cú pháp*: **double atof(*str_exp*)**
- *Ví dụ*: `double kq= atof('123.45');`

1.4.3. Hàm itoa

- Hàm thực hiện chuyển giá trị số nguyên sang dạng chuỗi và gán vào vùng nhớ mà con trỏ **st** đang trỏ đến. **st** là một con trỏ kiểu ký tự.
- *Cú pháp*: **char* itoa(int *value*, char **st*, int *radix*)**
- *Ví dụ*:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
void  main()
{
    int a=54325;
    char buffer[20];
```

```
    itoa(a,buffer,2);  // chuyển sang giá trị theo cơ số 2 (binary)
    printf("Binary value = %s\n", buffer); // Binary value = 1101010000110101

    itoa(a,buffer,10); // chuyển sang giá trị theo cơ số 10 (decimal)
    printf("Decimal value = %s\n", buffer); // Decimal value = 54325

    itoa(a,buffer,16); //chuyển sang giá trị theo cơ số 16 (Hexadecimal)
    printf("Hexadecimal value = %s\n", buffer); //Hexadecimal value = D435
}
```

Phần thực hành

2.1. Mục tiêu

Sinh viên phải biết cách nhập, xuất, lưu trữ trên mảng một chiều với các dữ liệu kiểu ký tự, xử lý các bài toán tìm kiếm, so sánh, cắt chuỗi, ...

2.2. Bài thực hành

(1)- Nhập chuỗi

(2)- Xuất chuỗi

(3)- Nhập vào 2 chuỗi, xuất chuỗi theo thứ tự từ điển

(4)- Đếm số ký tự ‘ a’ có trong chuỗi

(5)- Cắt khoảng trắng có trong chuỗi

(6)- Đếm khoảng trắng trong chuỗi .

(7)- Đếm số từ có trong chuỗi

(8)- Sắp xếp chuỗi tăng dần

(9)- Nhập 3 chuỗi, xuất chuỗi theo thứ tự từ điển.

2.3. Bài tập (sinh viên tự thực hiện)

(10)- Viết hàm nhập vào một mảng một chiều các số nguyên gồm n phần tử (0<n<100).

(11)- Viết hàm nhập vào một mảng một chiều các số thực gồm n phần tử (0<n<100).

(12)- Viết hàm xuất mảng số nguyên n phần tử vừa nhập ở trên .

(13)- Viết hàm xuất mảng số thực n phần tử vừa nhập ở trên .

(14)- Tính tổng các phần tử có trong mảng .

(15)- Tính tổng các phần tử chẵn có trong mảng .

(16)- Tính tổng các phần tử lẻ có trong mảng .

(17)- Tính tổng các phần tử nguyên tố có trong mảng .

(18)- Tìm phần tử chẵn đầu tiên có trong mảng

(19)- Tìm phần tử lẻ đầu tiên có trong mảng

(20)- Tìm phần tử nguyên tố đầu tiên có trong mảng

(21)- Tìm phần tử chẵn cuối cùng có trong mảng

(22)- Tìm phần tử chính phương cuối cùng có trong mảng

(23)- Tìm phần tử lớn nhất có trong mảng

(24)- Đếm số phần tử chẵn có trong mảng

(25)- Đếm số phần tử có giá trị là x có trong mảng

(26)- Đếm số phần tử lớn nhất có trong mảng

(27)- In ra vị trí của phần tử lớn nhất đầu tiên có trong mảng

(28)- In ra vị trí của phần tử có giá trị là x cuối cùng có trong mảng

(29)- Thêm một phần tử vào đầu mảng.

(30)- Thêm một phần tử vào cuối mảng.

(31)- Thêm một phần tử vào vị trí x trong mảng.

(32)- Xóa phần tử chẵn đầu tiên.

(33)- Xóa tất cả các phần tử lớn nhất trong mảng

(34)- Sắp xếp mảng tăng dần

(35)- Thêm một phần tử có giá trị là x vào trong mảng sao cho mảng vẫn có thứ tự tăng dần.

(36)- Đảo ngược mảng

(37)- Nhập chuỗi

(38)- Xuất chuỗi

(39)- Đếm số ký tự ‘ a’ có trong chuỗi

(40)- Cắt khoảng trắng có trong chuỗi

(41)- Đếm khoảng trắng trong chuỗi .

(42)- Đếm số từ có trong chuỗi

(43)- Sắp xếp chuỗi tăng dần

(44)- Nhập 3 chuỗi, xuất chuỗi theo thứ tự từ điển.