Bài tập ngôn ngữ lập trình C/C++

**Bài 1. Đọc mô tả về đoạn chương trình sau và trả lời câu hỏi:**

**[Mô tả chương trình]**

(1) Hàm ***match*** sau đây cho phép bạn khớp một xâu cụ thể với một xâu mẫu cho trước (p). Một xâu cụ thể chỉ bao gồm các ký tự alphabeta và dấu cách. Xâu mẫu bao gồm các ký tự alphabeta, dấu cách và các ký tự đại diện.

Các ký tự đại diện có thể:

‘?’ : thay thế cho một ký tự bất kỳ, thí dụ xâu mẫu a?b có thể khớp với xâu aCb, aeb…

‘\*’: thay thế cho số lượng bất kỳ các ký tự, thí dụ a\*c có thể khớp với xâu ac, abc, aEBDGc…

[tập các ký tự]: thay thế cho ký tự bất kỳ trong tập các ký tự được chỉ ra, thí dụ a[bcd]e có thể khớp với abe, ace, ade.

(2) ***p*** là con trỏ tới xâu mẫu kết thúc bằng ký tự null (\0).

(3) ***s*** là con trỏ tới xâu cần khớp và có chiều dài ***len*** ký tự, giả sử không có ký tự đại diện trong xâu cần khớp.

(4) ***len*** là chiều dài xâu cần khớp, thí dụ nếu ***s*** trỏ tới xâu “test case” và ***len*** = 4 thì xâu cần khớp sẽ là “test”.

**[Đoạn chương trình]**

#define TRUE 1

#define FALSE 0

int match(char \*p, char \*s, int len)

{

char c;

int i,matched;

c = \*p;

if (c == ’\0’)

return (len==0)?TRUE:FALSE;

if (c == ‘\*’) {

for (i = 0; ;i++)

if(match(p + 1,s + i,len – i)

return TRUE;

return FALSE;

}

if (len == 0) return FALSE;

if (c == ‘[‘) {

matched = FALSE;

for (i = 1;p[i] != ‘\0’;i++) {

if (p[i] == ‘]’)

return matched && match(p+i+1,s+1,len-1);

if (p[i] == \*s)

matched = TRUE;

}

return FALSE;

}

if (c == ‘?’)

return match (p + 1,s + 1, len – 1);

return (c==\*s) && match(p+1,s+1,len – 1);

}

**Câu hỏi con 1**

Chọn đáp án đúng điền vào dấu trong đoạn chương trình trên.

1. s[i] != ‘\0’ b) i < len
2. i <= len d) ( s[i] != ‘\0’) && (i < len)

**Câu hỏi con 2**

Chọn dòng nào có kết quả phù hợp với đoạn chương trình trên

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | match(“”, ”” , 0) | match (“?”,””,0) | match(“\*”,””,0) | match(“[]”,””,0) |
| a) | FALSE | FALSE | TRUE | TRUE |
| b) | TRUE | FALSE | FALSE | TRUE |
| c) | FALSE | TRUE | TRUE | FALSE |
| d) | TRUE | FALSE | TRUE | FALSE |

**Câu hỏi con 3**

Hàm match được gọi bao nhiêu lần khi câu lênh match(“??\*”,”test case”,4) được gọi

a)4 b) 5 c) 6 d) 10

**Câu hỏi con 4**

Hàm ***count\_match*** sau đây sẽ đếm tất cả các từ khớp với mẫu ***p* (** Một từ là tập hợp các ký tự liền nhau, thí dụ câu “this is a string” gồm 4 từ “this”, “is” , “a” , “string”).

int count\_matches(char \*p, char \*s)

{

char c, \*t;

int in\_word = FALSE, matches = 0;

do {

c = \*s;

if ((c>= ‘a’ && c <= ‘z’)||(c>= ‘A’ && c<= ‘Z’)) {

if (!in\_word){

in\_word = TRUE;

t = s;

}

}

else {

if (in\_word) {

in\_word = FALSE;

matches += match(p,t, );

}

}

s++;

} while (c);

return matches;

}

Chọn đáp án đúng điền vào chỗ trống trong đoạn chương trình trên

1. t – s b) s – t

c) strlen(s) d) strlen(t)

-------------------------------------------------------------

**Bài 2. Đọc mô tả về đoạn chương trình sau và trả lời câu hỏi:**

[**Mô tả chương trình**]

Hàm ***CountColors*** thực hiện quét dữ liệu ảnh trong hình chữ nhật kích thước 320x240 và đếm số màu trong đó. Mỗi điểm ảnh được lưu trong cấu trúc RGB như sau:

typedef struct {

unsigned char Red ; /\* Red brightness; 256 levels, from 0 to 255 \*/

unsigned char Green ;/\* Green brightness; 256 levels, from 0 to 255 \*/

unsigned char Blue ;/\* Blue brightness; 256 levels, from 0 to 255 \*/

} RGB ;

Dữ liệu ảnh được lưu trong mảng hai chiều kiểu RGB. Liên hệ giữa hình ảnh và hệ tọa độ được chỉ ra trong hình dưới đây.

RGB Image[ 240 ][ 320 ] ;



Fig. Hệ thống tọa độ cho dữ liệu ảnh

Sau khi dữ liệu ảnh được nạp vào mảng Image hàm ***CountColors*** được truyền các tham số mô tả khu vực cần đếm, kết quả trả về là số màu đếm được.

Định nghĩa hàm ***CountColors*** như sau

long int CountColors( int sx, int sy, int dx, int dy );

[**Đoạn chương trình**]

typedef struct {

unsigned char Red ;

unsigned char Green ;

unsigned char Blue ;

} RGB ;

#define Width 320

#define Height 240

#define TRUE 1

#define FALSE 0

RGB Image[ Height ][ Width ] ;

long int CountColors( int sx, int sy, int dx, int dy )

{

int w, h ;

int px, py ;

long int Colors ;

int Counted[ Height ][ Width ] ;

for( h = sy ; h <= dy ; h++ )

for( w = sx ; w <= dx ; w++ )

Counted[h][w] = FALSE ;

Colors = 0 ;

for( py = sy ; py <= dy ; py++ )

for( px = sx ; px <= dx ; px++ ) {

if ( Counted[py][px] == FALSE ) {

 ;

Counted[py][px] = TRUE ;

for( h = py ; h <= dy ; h++ )

for( w = (  ) ; w <= dx ; w++ ) {

if ( Counted[h][w] == FALSE ) {

if (( Image[py][px].Red == Image[h][w].Red )

&&( Image[py][px].Green == Image[h][w].Green )

&&( Image[py][px].Blue == Image[h][w].Blue ))

 ;

}

}

}

}

return Colors ;

}

**Câu hỏi con** : Điền phương án thích hợp vào chỗ trống.

Nhóm phương án cho “a”:

a) Colors = Counted[py][px] b) Colors = Counted[sy][sx]

c) Colors++ d) Colors−−

e) Counted[h][w] = FALSE f) Counted[h][w] = TRUE

Nhóm phương án cho “b”:

a) h == py ? 0 : dx b) h == py ? px + 1 : sx

c) h == py ? sx : px + 1 d) h == sy ? 0 : dx + 1

e) h == sy ? sx : px + 1

Nhóm phương án cho “c”:

a) Counted[dy][dx] = FALSE b) Counted[dy][dx] = TRUE

c) Counted[h][w] = FALSE d) Counted[h][w] = TRUE

e) Counted[py][px] = FALSE f) Counted[py][px] = TRUE

g) Counted[sy][sx] = FALSE h) Counted[sy][sx] = TRUE