

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC BÁCH KHOA ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN LẬP TRÌNH TÍNH TOÁN

ĐỀ TÀI:

Xây dựng ứng dụng thể hiện tam giác Pascal

Nhóm 15

Giảng viên hướng dẫn:
Phan Thanh Tào

Sinh viên thực hiện:
Nguyễn Hữu Tùng
Nguyễn Ngọc Tiệp
Lớp: 21T_DT

MỤC LỤC

0. MỞ ĐẦU	2
1. TỔNG QUAN ĐỀ TÀI	2
2. CƠ SỞ LÝ THUYẾT	2
2.1. Ý tưởng	2
2.2. Cơ sở lý thuyết	2
2.2.1. Quy hoạch động	2
3. TỔ CHỨC CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN	3
3.1. Phát biểu bài toán	3
3.2. Cấu trúc dữ liệu	3
3.3. Thuật toán	3
3.3.1. Sơ đồ khối thể hiện thuật toán	3
4. CHƯƠNG TRÌNH VÀ KẾT QUẢ	4
4.1. Tổ chức chương trình	4
4.2. Ngôn ngữ cài đặt	4
4.3. Kết quả	4
4.3.1. Giao diện chính của chương trình	4
4.3.2. Kết quả thực thi của chương trình	5
4.3.3. Nhận xét đánh giá	6
5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	6
5.1. Kết luận	6
5.2. Hướng phát triển	6
TÀI LIỆU THAM KHẢO	7
Phụ lục	8

0. MỞ ĐẦU:

Thông qua đề án, sinh viên có thể làm quen và nâng cao các kỹ năng:

1. Tổng hợp kiến thức, tra cứu, trích dẫn tài liệu
2. Phân tích, thiết kế, lựa chọn cấu trúc dữ liệu
3. Xây dựng thuật toán, lập trình
4. Giao tiếp, làm việc nhóm
5. Trình bày báo cáo, thuyết trình

Đối tượng cần nghiên cứu là bài toán Tam giác Pascal.

Phạm vi nghiên cứu nằm trong các kiến thức đã được học đối với sinh viên năm 1

Mục tiêu đề tài là in ra được dòng thứ N của tam giác, và in ra một tam giác Pascal có N dòng.

1. TỔNG QUAN ĐỀ TÀI:

Trong toán học, tam giác Pascal là một mảng tam giác của các hệ số nhị thức. Nó được đặt theo tên nhà toán học người Pháp Blaise Pascal.

Với ứng dụng này, ta có thể trình bày một tam giác Pascal với số lượng dòng lớn để có được cái nhìn bao quát hơn. Ngoài ra, ta còn có thể in ra một dòng cụ thể của nó.

2. Cơ sở lý thuyết:

2.1. Ý tưởng:

Qua quan sát ta nhận thấy dòng thứ N của tam giác Pascal là một danh sách các tổ hợp được liệt kê như sau:

$$NC_0, NC_1, \dots, NC_{N-1}, NC_N$$

Vậy nên mấu chốt bài toán là tìm được thuật toán tối ưu để tìm tổ hợp chập K của N phần tử.

2.2 Cơ sở lý thuyết:

2.2.1. Quy hoạch động:

Quy hoạch động (tiếng Anh: dynamic programming) là một phương pháp giảm thời gian chạy của các thuật toán thể hiện các tính chất của các bài toán con gối nhau và cấu trúc con tối ưu.

Ta có thể giải một bài toán với cấu trúc con tối ưu bằng một quy trình ba bước:

1. Chia bài toán thành các bài toán con nhỏ hơn.
2. Giải các bài toán này một cách tối ưu bằng cách sử dụng đệ quy quy trình ba bước này.
3. Sử dụng các kết quả tối ưu đó để xây dựng một lời giải tối ưu cho bài toán ban đầu.

Ta sử dụng phương pháp quy hoạch động để tìm tổ hợp chập K của N:

```

1. long long int tohop(int k, int n){
2. if(k==n || k==0) return 1;
3. return tohop(k,n-1) * n/(n-k);
4. }

```

3. Tổ chức cấu trúc dữ liệu và thuật toán:

3.1. Phát biểu bài toán:

Mô tả đầu vào (Input) và đầu ra (Output)

Đầu vào: nhập một số nguyên N để in ra Tam giác Pascal gồm N dòng, hoặc dòng thứ N cụ thể của nó.

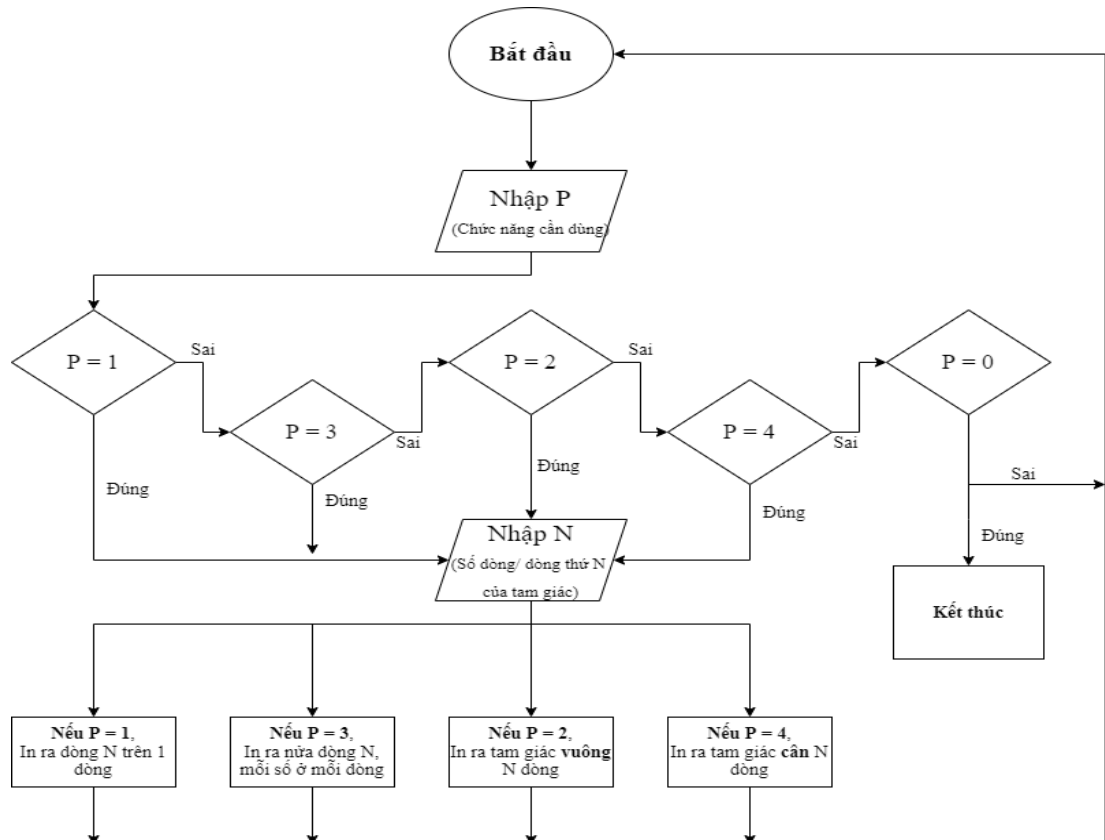
Đầu ra: Hiển thị Tam giác Pascal ra màn hình, hoặc dòng thứ N của nó.

3.2 Cấu trúc dữ liệu:

Chương trình chỉ sử dụng cấu trúc dữ liệu mảng.

3.3 Thuật toán:

3.3.1. Sơ đồ khối thể hiện thuật toán:



Hình 1. Sơ đồ khối

4. Chương trình và kết quả:

4.1. Tổ chức chương trình:

Chương trình chính: PascalTriangle.c

Các thư viện được sử dụng:

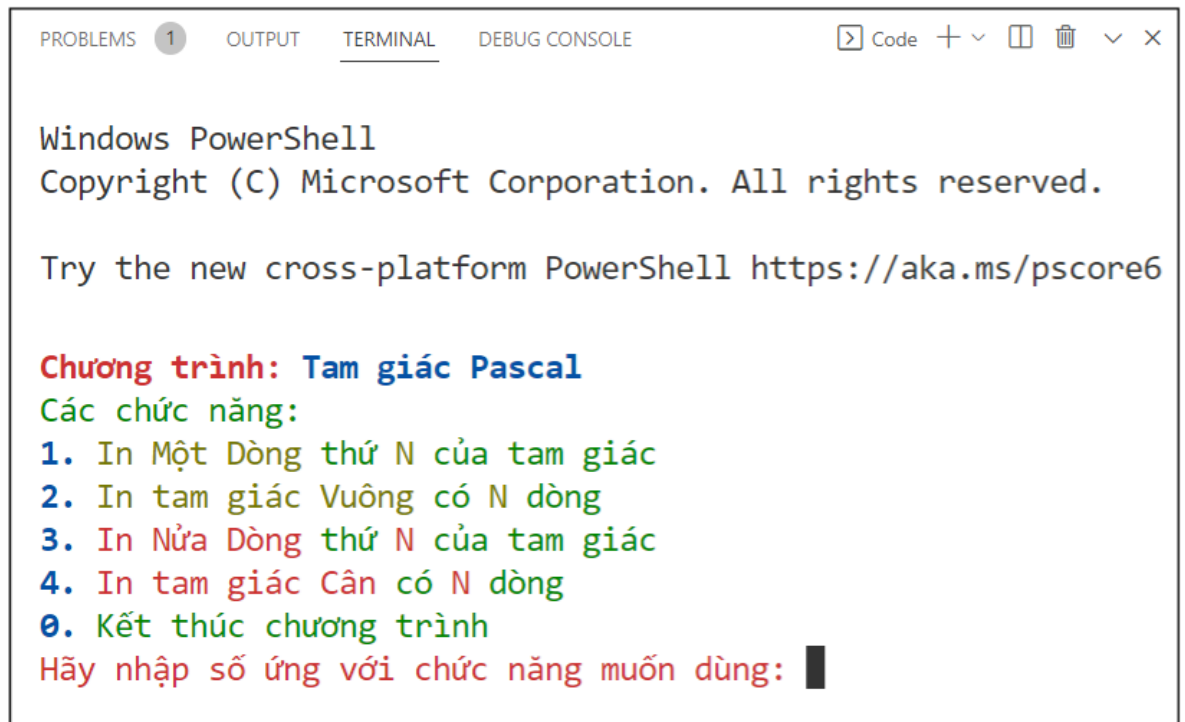
1. #include "ANSI-color-codes.h" // Thể hiện màu sắc trên terminal
2. #include <windows.h> // Xuất chuỗi unicode trên terminal
3. #include <stdio.h> // Thư viện C chuẩn

4.2. Ngôn ngữ cài đặt:

Chương trình sử dụng ngôn ngữ C.

4.3. Kết quả:

4.3.1. Giao diện chính của chương trình:

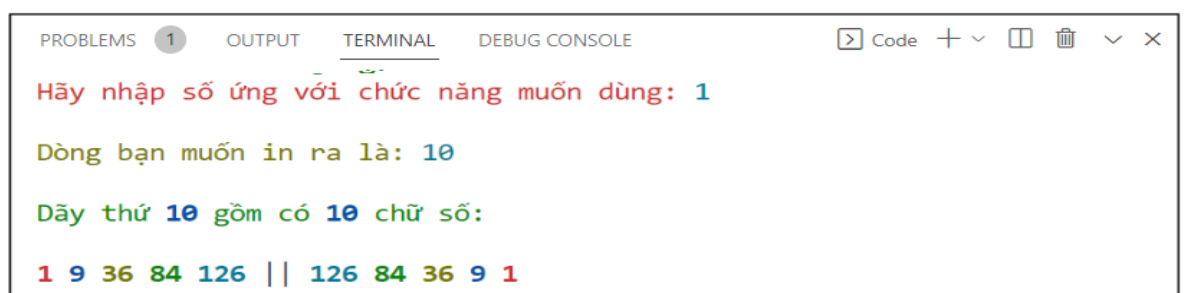


```
PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

Chương trình: Tam giác Pascal
Các chức năng:
1. In Một Dòng thứ N của tam giác
2. In tam giác Vuông có N dòng
3. In Nửa Dòng thứ N của tam giác
4. In tam giác Cân có N dòng
0. Kết thúc chương trình
Hãy nhập số ứng với chức năng muốn dùng: █
```

Hình 2. Màn hình chính



```
PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE
Hãy nhập số ứng với chức năng muốn dùng: 1

Dòng bạn muốn in ra là: 10

Dãy thứ 10 gồm có 10 chữ số:
1 9 36 84 126 || 126 84 36 9 1
```

Hình 3. In 1 dòng của tam giác

```
PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE
> Code + - [] [X]

Chương trình: Tam giác Pascal
Các chức năng:
1. In Một Dòng thứ N của tam giác
2. In tam giác Vuông có N dòng
3. In Nửa Dòng thứ N của tam giác
4. In tam giác Cân có N dòng
0. Kết thúc chương trình
Hãy nhập số ứng với chức năng muốn dùng: 4

Nhập số dòng muốn in của tam giác: 10

      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
```

Hình 4. In tam giác cân

```
PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE
> Code + - [] [X]

Chương trình: Tam giác Pascal
Các chức năng:
1. In Một Dòng thứ N của tam giác
2. In tam giác Vuông có N dòng
3. In Nửa Dòng thứ N của tam giác
4. In tam giác Cân có N dòng
0. Kết thúc chương trình
Hãy nhập số ứng với chức năng muốn dùng: 2

Nhập số dòng muốn in của tam giác: 10

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
```

Hình 5. In tam giác vuông

4.3.2. Kết quả thực thi của chương trình:

Kết quả khi thực thi chương trình:

1. Chương trình dựa trên chức năng đã chọn, và số nguyên N để thực hiện.

2. In ra 1 dòng, hoặc 1 tam giác hoàn chỉnh tùy chức năng
3. Nhấn số 0 để thoát khỏi chương trình

4.3.3. Nhận xét đánh giá:

Kết quả mong đợi:

1. Hiểu rõ và hiểu được khái niệm, tính chất của mảng và thuật toán quy hoạch động.
2. Áp dụng được ngôn ngữ C để tạo ra chương trình .
3. Hiểu và tạo một số hàm, thư viện để thêm màu sắc vào trong chương trình.

Kết quả đạt được:

1. Củng cố và mở rộng kiến thức về ngôn ngữ C
2. Hiểu rõ các bước tạo ra một chương trình chạy trên terminal.

Sau khi hoàn thành, chúng em đã đạt được những kết quả và gần so khớp với kết quả như mong đợi, tuy nhiên về phần thuật toán, kiến thức cũng như kinh nghiệm chúng em vẫn còn nhiều hạn chế và thiếu sót trong quá trình nghiên cứu. Vì thế, cần nhiều thời gian tìm tòi, học hỏi và nghiên cứu nhiều hơn.

5. Kết luận và hướng phát triển:

5.1. Kết luận:

Trong quá trình nghiên cứu về đề tài “Xây dựng ứng dụng thể hiện tam giác Pascal” chúng em đã đạt được nhiều kiến thức cả về mặt lý thuyết lẫn kinh nghiệm như:

1. Hiểu rõ ràng về cách xây dựng 1 chương trình chạy trên Terminal
2. Có thêm kinh nghiệm trong việc làm việc nhóm
3. Xây dựng kỹ năng giải quyết vấn đề được đặt ra

5.2. Hướng phát triển:

Sau các kết quả đã đạt được từ quá trình tìm hiểu, nghiên cứu và làm việc, chúng em có một số hướng phát triển cho đề án như sau:

1. Cải thiện độ chính xác của các thuật toán
2. Hoàn thiện hơn về mặt giao diện và cách sắp xếp các hàm
3. Lược bỏ, làm gọn, đơn giản hóa một số chức năng

Tài liệu tham khảo

1. GeeksForGeeks, Phương pháp tìm dòng thứ Nth trong tam giác Pascal
<https://www.geeksforgeeks.org/find-the-nth-row-in-pascals-triangle/>
2. Stackoverflow, In màu trên Terminal
<https://stackoverflow.com/questions/3219393/stdlib-and-colored-output-in-c>
3. W3resource, Thuật toán khoảng cách Tam giác cân
<https://www.w3resource.com/c-programming-exercises/for-loop/c-for-loop-exercises-33.php>

Phụ lục

PascalTriangle.c

```
1. #include "ANSI-color-codes.h"
2. #include <stdio.h>
3. #include <windows.h>
4.
5. void menu();
6. void keepUsing();
7. void colorShift(int j);
8.
9. //nCk
10. long long int tohop(int k, int n){
11.     if(k==n || k==0) return 1;
12.     return tohop(k,n-1) * n/(n-k);
13. }
14. }
15.
16. void printNthLineStraight(){
17.     int n;
18.     printf(YEL "\nDòng bạn muốn in ra là: " CYN);
19.     scanf("%d", &n); printf(RESET);
20.     n -= 1;
21.
22.     if(n <= -1){
23.         printf(RED "\nSố dòng phải bắt đầu từ số " CYN "1 " RED "trở lên \n\n"
24.         RESET);
25.         keepUsing(); return;
26.     }
27.
28.     if(n == 0){
29.         printf(GRN "\nChỉ có 1 số: " CYN "1 \n\n" RESET);
30.         keepUsing(); return;
31.     }
32.     printf(GRN "\nDãy thứ " BBLU); printf("%d",n+1); printf(GRN " gồm có " BBLU
33.     "%d",n+1); printf(GRN " chữ số:\n\n" RESET);
34.     int colorState = 0;
35.     long long int keepNumber[100000];
36.     int pos = (n+1)/2;
37.     if(n % 2 == 0){ // n = 2;4;6
38.         for (int i = 0; i <= pos; i++)
39.         {
40.             colorShift(colorState);
41.             long long temp = tohop(i,n);
42.             if(i == pos){
43.                 printf(RESET "||");
44.                 colorShift(colorState);
45.                 printf(" %lli ",temp);
46.                 printf(RESET "|| ");
47.                 colorState--;
48.             }else{
49.                 printf("%lli ",temp);
50.             }
51.             keepNumber[pos+ (i+1)] = temp;
52.             colorState++;
53.         }
54.         for(int i = (n); i > pos; i--){
55.             colorState--;
56.             colorShift(colorState);
57.             printf("%lli ",keepNumber[i]);
```

```

58.     }
59.     printf("\n");
60.     printf(RESET);
61.     //keepUsing(); return;
62. }else{ // n = 1;3;5;7
63.
64.     for (int i = 0; i < pos; i++)
65.     {
66.         colorShift(colorState);
67.         long long temp = tohop(i,n);
68.         printf("%lli ",temp);
69.         keepNumber[pos+ (i+1)] = temp;
70.         colorState++;
71.         if(i == pos-1){
72.             printf(RESET "|| ");
73.         }
74.     }
75.     for(int i = (n+1); i > pos; i--){
76.         colorState--;
77.         colorShift(colorState);
78.         printf("%lli ",keepNumber[i]);
79.     }
80.     printf("\n");
81.     printf(RESET);
82.     //keepUsing(); return;
83. }
84. menu();
85. }
86. //=====
87. void printNthLine(){
88.     int n;
89.     printf(YEL "\nDòng bạn muốn in ra là: " CYN);
90.     scanf("%d", &n); printf(RESET);
91.     n -= 1;
92.     if(n <= -1){
93.         printf(RED "\nSố dòng phải bắt đầu từ số " CYN "1 " RED "trở lên \n\n"
RESET);
94.         keepUsing(); return;
95.     }
96.     if(n == 0){
97.         printf(GRN "\nChỉ có 1 số: " CYN "1 \n\n" RESET);
98.         keepUsing(); return;
99.     }
100.
101.     printf(GRN "\nDãy thứ " BBLU); printf("%d",n+1); printf(GRN " gồm có " BBLU
"%d",n+1); printf(GRN" chữ số\n" RESET);
102.     int pos = (n + 1)/2;
103.     if(n % 2 == 0){ // n = 2;4;6
104.         for (int i = 0; i <= pos; i++)
105.         {
106.             printf(GRN "Số thứ " BBLU); printf("%d: ",i+1); printf(GRN ":" RED
" %lli\n",tohop(i,n));
107.         }
108.         printf(GRN "Bỏ đi số chính giữa thì nửa sau giống nửa trước, nhưng đảo
chiều\n");
109.         printf(RESET);
110.         //keepUsing(); return;
111.     }else{ // n = 1;3;5;7
112.
113.         for (int i = 0; i < pos; i++)
114.         {
115.             printf(GRN "Số thứ " BBLU); printf("%d: ",i+1); printf(GRN ":" RED
" %lli\n",tohop(i,n));
116.         }
117.         printf(GRN "Nửa sau giống nửa trước, nhưng đảo chiều\n");

```

```

118.         printf(RESET);
119.         //keepUsing(); return;
120.     }
121.     menu();
122. }
123.
124. //Cau 2 https://www.w3resource.com/c-programming-exercises/for-loop/c-for-loop-exercises-33.php=====
125.
126. void printPascalTriangle(int option){
127.     int no_row,blk,i,j;
128.     long long int c = 1;
129.     printf(YEL "\nNhập số dòng muốn in của tam giác: " CYN);
130.     scanf("%d", &no_row); printf(RESET);
131.     if(no_row >= 27){
132.         printf(RED "Từ dòng 27 trở đi sẽ ko đủ chỗ hiển thị, bạn có muốn tiếp
tục? (1 = YES, 0 = NO) " CYN);
133.         int temp; scanf("%d", &temp); printf(RESET);
134.         if(temp == 0){
135.             menu(); return;
136.         }
137.     }
138.
139.     for(i=0;i<no_row;i++)
140.     {
141.         if(option){
142.             for(blk=1;blk < no_row-i;blk++){
143.                 if(no_row < 5){
144.                     printf(" ");
145.                 }else if(no_row < 14){
146.                     printf(" ");
147.                 }else if(no_row < 21){
148.                     printf(" ");
149.                 }else if(no_row >= 21){
150.                     printf(" ");
151.                 }
152.             }
153.         }
154.         int pos = -1;
155.         if(i % 2 != 0){
156.             pos = 0;
157.         }
158.         for(j=0;j<=i;j++)
159.         {
160.             if(i % 2 == 0){ //dòng có số chính giữa , i = 0;2;4
161.                 if(j <= i/2){ //set màu tới số chính giữa
162.                     colorShift(j);
163.                     pos++;
164.                 }else{
165.                     pos--;
166.                     colorShift(pos);
167.                 }
168.             }else{//ko có số giữa, i = 1;3;5
169.                 if(j < (i+1)/2){ //set màu về trái
170.                     colorShift(j);
171.                     pos++;
172.                 }else{
173.                     pos--;
174.                     colorShift(pos);
175.                 }
176.             }
177.
178.             c = tohop(j,i);
179.             if(j == 0){

```

```

180.         printf("%d",c);
181.     }else{
182.         if(no_row < 5){
183.             printf("% 2d",c);
184.         }else if(no_row < 14){
185.             printf("% 4d",c);
186.         }else if(no_row < 21){
187.             printf("% 6d",c);
188.         }else if(no_row >= 21){
189.             printf("% 8d",c);
190.         }
191.     }
192. }
193.     printf("\n");
194. }
195.     printf(RESET); menu();
196. }
197. //=====
198. int option;
199. void main(){
200.     SetConsoleOutputCP(CP_UTF8);
201.
202.     menu();
203.     while(option){
204.         if(option == 1){
205.             printNthLineStraight();
206.         }else if(option == 2){
207.             printPascalTriangle(0);
208.         }else if(option == 3){
209.             printNthLine();
210.         }else if(option == 4){
211.             printPascalTriangle(1);
212.         }else{
213.             printf(BMAG "\nYêu cầu không hợp lệ, lựa chọn chỉ gồm 5 số " BYEL
214. "1-2-3-4-0 \n"RESET);
215.             menu();
216.         }
217.     }
218.     printf(UYEL "Cảm ơn bạn đã dùng phần mềm"RESET);
219.
220. void menu(){
221.     //printf(RED "If" BLU "you" YEL "are" GRN "bored" CYN "do" MAG "this! \n"
222. RESET);
223.     //printf(BRED "If" BBLU "you" BYEL "are" BGRN "bored" BCYN "do" BMAG
224. "this! \n" RESET);
225.     //printf(URED "If" UBLU "you" UYEL "are" UGRN "bored" UCYN "do" UMAG
226. "this! \n" RESET);
227.     printf("\n");
228.     printf(BRED "Chương trình:" BBLU " Tam giác Pascal\n" RESET);
229.     printf(GRN "Các chức năng:\n");
230.     printf(BBLU "1." YEL " In Một Dòng" GRN " thứ " YEL "N " GRN "của tam
231. giác\n");
232.     printf(BBLU "2." YEL " In tam giác Vuông" GRN " có " YEL "N " GRN
233. "dòng\n");
234.     printf(BBLU "3." RED " In Nửa Dòng" GRN " thứ " RED "N " GRN "của tam
235. giác\n");
236.     printf(BBLU "4." RED " In tam giác Cân" GRN " có " RED "N " GRN "dòng\n");
237.     printf(BBLU "0." GRN " Kết thúc chương trình\n" RESET);
238.     printf(RED "Hãy nhập số ứng với chức năng muốn dùng: " CYN);
239.     scanf("%d", &option);
240.     printf(RESET);
241. }
242.
243. void keepUsing(){

```

```

238.     printf(YEL "Bạn có muốn dùng tiếp không (1 = Yes, 0 = No): " CYN);
239.     int temp;
240.     scanf("%d", &temp);
241.     if(temp){
242.         menu();
243.     }else{
244.         printf(URED "Cảm ơn bạn đã dùng phần mềm"RESET);
245.         exit(1);
246.     }
247. }
248. void colorShift(int j){
249.     if(j % 6 == 0){
250.         printf(BRED);
251.     }else if(j % 6 == 1){
252.         printf(BBLU);
253.     }else if(j % 6 == 2){
254.         printf(BYEL);
255.     }else if(j % 6 == 3){
256.         printf(BGRN);
257.     }else if(j % 6 == 4){
258.         printf(BCYN);
259.     }else if(j % 6 == 5){
260.         printf(BMAG);
261.     }
262. }
263.

```

ANSI-color-codes.h

```

1. //Regular text
2. #define BLK "\e[0;30m"
3. #define RED "\e[0;31m"
4. #define GRN "\e[0;32m"
5. #define YEL "\e[0;33m"
6. #define BLU "\e[0;34m"
7. #define MAG "\e[0;35m"
8. #define CYN "\e[0;36m"
9. #define WHT "\e[0;37m"
10.
11. //Regular bold text
12. #define BBLK "\e[1;30m"
13. #define BRED "\e[1;31m"
14. #define BGRN "\e[1;32m"
15. #define BYEL "\e[1;33m"
16. #define BBLU "\e[1;34m"
17. #define BMAG "\e[1;35m"
18. #define BCYN "\e[1;36m"
19. #define BWHT "\e[1;37m"
20.
21. //Regular underline text
22. #define UBLK "\e[4;30m"
23. #define URED "\e[4;31m"
24. #define UGRN "\e[4;32m"
25. #define UYEL "\e[4;33m"
26. #define UBLU "\e[4;34m"
27. #define UMAG "\e[4;35m"
28. #define UCYN "\e[4;36m"
29. #define UWHT "\e[4;37m"
30.
31. //Regular background

```

```

32. #define BLKB "\e[40m"
33. #define REDB "\e[41m"
34. #define GRNB "\e[42m"
35. #define YELB "\e[43m"
36. #define BLUB "\e[44m"
37. #define MAGB "\e[45m"
38. #define CYNB "\e[46m"
39. #define WHTB "\e[47m"
40.
41. //High intensty background
42. #define BLKHB "\e[0;100m"
43. #define REDHB "\e[0;101m"
44. #define GRNHB "\e[0;102m"
45. #define YELHB "\e[0;103m"
46. #define BLUHB "\e[0;104m"
47. #define MAGHB "\e[0;105m"
48. #define CYNHB "\e[0;106m"
49. #define WHTHB "\e[0;107m"
50.
51. //High intensty text
52. #define HBLK "\e[0;90m"
53. #define HRED "\e[0;91m"
54. #define HGRN "\e[0;92m"
55. #define HYEL "\e[0;93m"
56. #define HBLU "\e[0;94m"
57. #define HMAG "\e[0;95m"
58. #define HCYN "\e[0;96m"
59. #define HWHT "\e[0;97m"
60.
61. //Bold high intensity text
62. #define BHBLK "\e[1;90m"
63. #define BHRED "\e[1;91m"
64. #define BHGRN "\e[1;92m"
65. #define BHYEL "\e[1;93m"
66. #define BHBLU "\e[1;94m"
67. #define BHMAG "\e[1;95m"
68. #define BHCYN "\e[1;96m"
69. #define BHWHT "\e[1;97m"
70.
71. //Reset
72. #define RESET "\e[0m"
73. #define CRESET "\e[0m"
74. #define COLOR_RESET "\e[0m"
75.

```