

SoftwareLab3

Generated by Doxygen 1.8.6

Sun Feb 1 2015 20:56:22

Contents

1	File Index	1
1.1	File List	1
2	File Documentation	3
2.1	new.c File Reference	3
2.1.1	Macro Definition Documentation	3
2.1.1.1	number	3
2.1.2	Function Documentation	3
2.1.2.1	create	3
2.1.2.2	main	4
2.1.2.3	sets	4
2.2	new2.c File Reference	4
2.2.1	Macro Definition Documentation	5
2.2.1.1	number	5
2.2.2	Function Documentation	5
2.2.2.1	compare	5
2.2.2.2	create	5
2.2.2.3	main	6
2.2.2.4	sets	6
2.2.3	Variable Documentation	6
2.2.3.1	base	6
2.2.3.2	count	6
2.3	new3.c File Reference	6
2.3.1	Macro Definition Documentation	7
2.3.1.1	number	7
2.3.2	Function Documentation	7
2.3.2.1	compare	7
2.3.2.2	create	7
2.3.2.3	main	8
2.3.2.4	sets	8
2.3.3	Variable Documentation	8

2.3.3.1	base	8
2.3.3.2	count	9
2.4	new_astyle.c File Reference	9
2.4.1	Macro Definition Documentation	9
2.4.1.1	number	9
2.4.2	Function Documentation	9
2.4.2.1	create	9
2.4.2.2	main	10
2.4.2.3	sets	10
	Index	11

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

new.c	3
new2.c	4
new3.c	6
new_astyle.c	9

Chapter 2

File Documentation

2.1 new.c File Reference

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
```

Macros

- `#define number 50`

Functions

- void `sets` (int a[], int K)
- void `create` ()
- int `main` (void)

2.1.1 Macro Definition Documentation

2.1.1.1 `#define number 50`

2.1.2 Function Documentation

2.1.2.1 void `create` ()

```
36 {
37     int b[4];
38     int *a;
39     a = (int*)malloc(number*sizeof(int));
40     int i, j, k, l, K;
41     char c;
42     for(i=0; i<number; i++)
43         scanf("%d%c", &a[i], &c);
44     //srand ( time(NULL) );
45     //for (i=0; i<number; i++)
46     //{
47     //    a[i] = (int)pow(-1, rand()%2) * (rand() % number + 1);
48     //}
49     // for (i=0; i<number; i++)
50     //     fprintf(f, "%d\n", a[i]);
51     scanf("%d", &K);
52     for(i=0; i<number; i++)
53     {
54         b[0]=a[i];
55         for(j=i+1; j<number; j++)
```

```

56     {
57         b[1]=a[j];
58         for(k=j+1;k<number;k++)
59         {
60             b[2]=a[k];
61             for(l=k+1;l<number;l++)
62             {
63                 b[3]=a[l];
64                 //printf("%d %d %d %d\n",b[0],b[1],b[2],b[3]);
65                 sets(b,K);
66             }
67         }
68     }
69 }
70
71 }

```

2.1.2.2 int main (void)

```

74 {
75     create();
76     return 0;
77 }

```

2.1.2.3 void sets (int a[], int K)

```

6 {
7     int i,j,k,t,prod,temp[2];
8     for(i=0;i<3;i++)
9     {
10         for(j=i+1;j<4;j++)
11         {
12             prod=1;
13             t=0;
14             for(k=0;k<4;k++)
15             {
16                 if(k!=i&&k!=j)
17                 {
18                     prod=prod*a[k];
19                     temp[t]=k;
20                     t++;
21                 }
22             }
23             if(a[i]+a[j]+prod<= K)
24             {
25                 //printf("%d,%d,%d,%d\n",a[i],a[j],a[temp[0]],a[temp[1]]);
26                 //printf("%d,%d,%d,%d\n",a[i],a[j],a[temp[1]],a[temp[0]]);
27                 //printf("%d,%d,%d,%d\n",a[j],a[i],a[temp[0]],a[temp[1]]);
28                 //printf("%d,%d,%d,%d\n",a[j],a[i],a[temp[1]],a[temp[0]]);
29             }
30         }
31     }
32 }

```

2.2 new2.c File Reference

```

#include <stdio.h>
#include <math.h>
#include <stdlib.h>

```

Macros

- #define `number` 50

Functions

- int `compare` (int arr[4])

- void `sets` (int a[], int K)
- void `create` ()
- int `main` (void)

Variables

- int `count` =0
- int `base` [4000000][4]

2.2.1 Macro Definition Documentation

2.2.1.1 #define number 50

2.2.2 Function Documentation

2.2.2.1 int compare (int arr[4])

```

9 {
10     int i;
11
12     for(i=0; i<count; i++)
13     {
14         if(base[i][0]==arr[0]&&base[i][1]==arr[1]&&base[i][2]==arr[2]&&
base[i][3]==arr[3])
15             return 1;
16     }
17
18     for(i=0; i<4; i++)
19         base[count][i]=arr[i];
20     count++;
21     return 0;
22
23 }
```

2.2.2.2 void create ()

```

56 {
57     int b[4],action;
58     int *a;
59     a = (int*)malloc(number*sizeof(int));
60     int i,j,k,l,K;
61     char c;
62     //for(i=0;i<number;i++)
63     //    scanf("%d%c",&a[i],&c);
64     srand (time(NULL));
65     for(i=0; i<number; i++)
66     {
67         a[i]= (int)pow(-1,rand()%2)*(rand() % number + 1);
68     }
69     for(i=0; i<number; i++)
70         printf("%d,",a[i]);
71     scanf("%d",&K);
72     for(i=0; i<number; i++)
73     {
74         b[0]=a[i];
75         for(j=i+1; j<number; j++)
76         {
77             b[1]=a[j];
78             for(k=j+1; k<number; k++)
79             {
80                 b[2]=a[k];
81                 for(l=k+1; l<number; l++)
82                 {
83                     b[3]=a[l];
84                     //printf("%d %d %d %d\n",b[0],b[1],b[2],b[3]);
85                     action = compare(b);
86                     if(action==0)
87                         sets(b,K);
88                 }
89             }
90         }
91     }
```

```

92
93 }

```

2.2.2.3 int main (void)

```

96 {
97     int i, j;
98     create();
99     //printf("stored");
100    //for (i=0; i<count; i++)
101    //{
102        //    for (j=0; j<4; j++)
103            //        printf("%d, ", base[i][j]);
104        //    printf("\n");
105    //}
106    return 0;
107 }

```

2.2.2.4 void sets (int a[], int K)

```

26 {
27     int i, j, k, t, prod, temp[2];
28     for (i=0; i<3; i++)
29     {
30         for (j=i+1; j<4; j++)
31         {
32             prod=1;
33             t=0;
34             for (k=0; k<4; k++)
35             {
36                 if (k!=i&& k!=j)
37                 {
38                     prod=prod*a[k];
39                     temp[t]=k;
40                     t++;
41                 }
42             }
43             if (a[i]+a[j]+prod<= K)
44             {
45                 //printf("%d, %d, %d, %d\n", a[i], a[j], a[temp[0]], a[temp[1]]);
46                 //printf("%d, %d, %d, %d\n", a[i], a[j], a[temp[1]], a[temp[0]]);
47                 //printf("%d, %d, %d, %d\n", a[j], a[i], a[temp[0]], a[temp[1]]);
48                 //printf("%d, %d, %d, %d\n", a[j], a[i], a[temp[1]], a[temp[0]]);
49             }
50         }
51     }
52 }

```

2.2.3 Variable Documentation

2.2.3.1 int base[4000000][4]

2.2.3.2 int count =0

2.3 new3.c File Reference

```

#include <stdio.h>
#include <math.h>
#include <stdlib.h>

```

Macros

- #define [number](#) 50

Functions

- int `compare` (int arr[4])
- void `sets` (int a[], int K)
- void `create` ()
- int `main` (void)

Variables

- int `count` =0
- int `base` [4000000][4]

2.3.1 Macro Definition Documentation

2.3.1.1 #define number 50

2.3.2 Function Documentation

2.3.2.1 int compare (int arr[4])

```

9 {
10     int i;
11
12     for(i=0; i<count; i++)
13     {
14         if(base[i][0]==arr[0]&&base[i][1]==arr[1]&&base[i][2]==arr[2]&&
base[i][3]==arr[3])
15             return 1;
16     }
17
18     for(i=0; i<4; i++)
19         base[count][i]=arr[i];
20     count++;
21     return 0;
22 }
23 }
```

2.3.2.2 void create ()

```

75 {
76     int b[4],action;
77     int *a;
78     a = (int*)malloc(number*sizeof(int));
79     int i,j,k,l,K;
80     char c;
81     for(i=0; i<number; i++)
82         scanf("%d%c",&a[i],&c);
83     //srand (time(NULL));
84     //for(i=0;i<number;i++)
85     // {
86     //     a[i]= (int)pow(-1,rand()%2)*(rand() % number + 1);
87     // }
88     // for(i=0;i<number;i++)
89     //     fprintf(f,"%d\n",a[i]);
90     scanf("%d",&K);
91     for(i=0; i<number; i++)
92     {
93         b[0]=a[i];
94         for(j=i+1; j<number; j++)
95         {
96             b[1]=a[j];
97             for(k=j+1; k<number; k++)
98             {
99                 b[2]=a[k];
100                 for(l=k+1; l<number; l++)
101                 {
102                     b[3]=a[l];
103                     //printf("%d %d %d %d\n",b[0],b[1],b[2],b[3]);
104                     action = compare(b);
105                     if(action==0)
```

```

106             sets(b,K);
107         }
108     }
109 }
110 }
111
112 }

```

2.3.2.3 int main (void)

```

115 {
116     int i,j;
117     create();
118     //printf("stored");
119     //for(i=0;i<count;i++)
120     //{
121         // for(j=0;j<4;j++)
122         //     printf("%d",base[i][j]);
123     // printf("\n");
124     //}
125     return 0;
126 }

```

2.3.2.4 void sets (int a[], int K)

```

26 {
27     if(a[0]+a[1]+a[2]*a[3]<=K)
28     {
29         //printf("%d,%d,%d,%d\n",a[0],a[1],a[2],a[3]);
30         //printf("%d,%d,%d,%d\n",a[0],a[1],a[3],a[2]);
31         //printf("%d,%d,%d,%d\n",a[1],a[0],a[2],a[3]);
32         //printf("%d,%d,%d,%d\n",a[1],a[0],a[3],a[2]);
33     }
34     if(a[0]+a[2]+a[1]*a[3]<=K)
35     {
36         //printf("%d,%d,%d,%d\n",a[0],a[2],a[1],a[3]);
37         //printf("%d,%d,%d,%d\n",a[0],a[2],a[3],a[1]);
38         //printf("%d,%d,%d,%d\n",a[2],a[0],a[1],a[3]);
39         //printf("%d,%d,%d,%d\n",a[2],a[0],a[3],a[1]);
40     }
41     if(a[0]+a[3]+a[1]*a[2]<=K)
42     {
43         //printf("%d,%d,%d,%d\n",a[0],a[3],a[1],a[2]);
44         //printf("%d,%d,%d,%d\n",a[3],a[0],a[2],a[1]);
45         //printf("%d,%d,%d,%d\n",a[3],a[0],a[1],a[2]);
46         //printf("%d,%d,%d,%d\n",a[0],a[3],a[2],a[1]);
47     }
48     if(a[1]+a[2]+a[0]*a[3]<=K)
49     {
50         //printf("%d,%d,%d,%d\n",a[1],a[2],a[0],a[3]);
51         //printf("%d,%d,%d,%d\n",a[1],a[2],a[3],a[0]);
52         //printf("%d,%d,%d,%d\n",a[2],a[1],a[0],a[3]);
53         //printf("%d,%d,%d,%d\n",a[2],a[1],a[3],a[0]);
54     }
55     if(a[1]+a[3]+a[0]*a[2]<=K)
56     {
57         //printf("%d,%d,%d,%d\n",a[3],a[1],a[2],a[0]);
58         //printf("%d,%d,%d,%d\n",a[3],a[1],a[2],a[0]);
59         //printf("%d,%d,%d,%d\n",a[1],a[3],a[0],a[2]);
60         //printf("%d,%d,%d,%d\n",a[1],a[3],a[0],a[2]);
61     }
62     if(a[2]+a[3]+a[0]*a[1]<=K)
63     {
64         //printf("%d,%d,%d,%d\n",a[2],a[3],a[0],a[1]);
65         //printf("%d,%d,%d,%d\n",a[2],a[3],a[1],a[0]);
66         //printf("%d,%d,%d,%d\n",a[3],a[2],a[1],a[0]);
67         //printf("%d,%d,%d,%d\n",a[3],a[2],a[0],a[1]);
68     }
69 }
70 }

```

2.3.3 Variable Documentation

2.3.3.1 int base[4000000][4]

2.3.3.2 int count =0

2.4 new_astyle.c File Reference

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
```

Macros

- #define [number](#) 50

Functions

- void [sets](#) (int a[], int K)
- void [create](#) ()
- int [main](#) (void)

2.4.1 Macro Definition Documentation

2.4.1.1 #define number 50

2.4.2 Function Documentation

2.4.2.1 void create ()

```
36 {
37     int b[4];
38     int *a;
39     a = (int*)malloc(number*sizeof(int));
40     int i,j,k,l,K;
41     char c;
42     for(i=0; i<number; i++)
43         scanf("%d%c",&a[i],&c);
44     //srand ( time(NULL) );
45     //for(i=0;i<number;i++)
46     //{
47     //    a[i]= (int)pow(-1,rand()%2)*(rand() % number + 1);
48     //}
49     // for(i=0;i<number;i++)
50     //     fprintf(f,"%d\n",a[i]);
51     scanf("%d",&K);
52     for(i=0; i<number; i++)
53     {
54         b[0]=a[i];
55         for(j=i+1; j<number; j++)
56         {
57             b[1]=a[j];
58             for(k=j+1; k<number; k++)
59             {
60                 b[2]=a[k];
61                 for(l=k+1; l<number; l++)
62                 {
63                     b[3]=a[l];
64                     //printf("%d %d %d %d\n",b[0],b[1],b[2],b[3]);
65                     sets(b,K);
66                 }
67             }
68         }
69     }
70 }
71 }
```

2.4.2.2 int main (void)

```
74 {  
75     create();  
76     return 0;  
77 }
```

2.4.2.3 void sets (int a[], int K)

```
6 {  
7     int i,j,k,t,prod,temp[2];  
8     for(i=0; i<3; i++)  
9     {  
10         for(j=i+1; j<4; j++)  
11         {  
12             prod=1;  
13             t=0;  
14             for(k=0; k<4; k++)  
15             {  
16                 if(k!=i&&k!=j)  
17                 {  
18                     prod=prod*a[k];  
19                     temp[t]=k;  
20                     t++;  
21                 }  
22             }  
23             if(a[i]+a[j]+prod<= K)  
24             {  
25                 //printf("%d,%d,%d,%d\n",a[i],a[j],a[temp[0]],a[temp[1]]);  
26                 //printf("%d,%d,%d,%d\n",a[i],a[j],a[temp[1]],a[temp[0]]);  
27                 //printf("%d,%d,%d,%d\n",a[j],a[i],a[temp[0]],a[temp[1]]);  
28                 //printf("%d,%d,%d,%d\n",a[j],a[i],a[temp[1]],a[temp[0]]);  
29             }  
30         }  
31     }  
32 }
```

Index

base
 new2.c, [6](#)
 new3.c, [8](#)

compare
 new2.c, [5](#)
 new3.c, [7](#)

count
 new2.c, [6](#)
 new3.c, [8](#)

create
 new.c, [3](#)
 new2.c, [5](#)
 new3.c, [7](#)
 new_astyle.c, [9](#)

main
 new.c, [4](#)
 new2.c, [6](#)
 new3.c, [8](#)
 new_astyle.c, [9](#)

new.c, [3](#)
 create, [3](#)
 main, [4](#)
 number, [3](#)
 sets, [4](#)

new2.c, [4](#)
 base, [6](#)
 compare, [5](#)
 count, [6](#)
 create, [5](#)
 main, [6](#)
 number, [5](#)
 sets, [6](#)

new3.c, [6](#)
 base, [8](#)
 compare, [7](#)
 count, [8](#)
 create, [7](#)
 main, [8](#)
 number, [7](#)
 sets, [8](#)

new_astyle.c, [9](#)
 create, [9](#)
 main, [9](#)
 number, [9](#)
 sets, [10](#)

number
 new.c, [3](#)
 new2.c, [5](#)
 new3.c, [7](#)
 new_astyle.c, [9](#)

sets
 new.c, [4](#)
 new2.c, [6](#)
 new3.c, [8](#)
 new_astyle.c, [10](#)