

主加法標準形への展開のソースコードを，ソースコード 1 に示す．．．

ソースコード 1: hoge

```
1 #include<stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 #define N 100
6
7 void usage(char *f){ //入力の確認
8     int i;
9
10    for(i=0 ;i<N ;i++){
11        if(f[i] == 0)
12            break;
13        switch(f[i]){
14            case 'x':
15            case 'y':
16            case 'z':
17            case '+':
18                break;
19            case '*':
20                switch(f[i-1]){
21                    case 'x':
22                    case 'y':
23                    case 'z':
24                        break;
25                    default:
26                        printf("error\n");
27                        exit(1);
28                }
29                break;
30            default:
31                printf("error\n");
32                exit(1);
33        }
34    }
35 }
36
37 char and(char a ,char b){
38     if(a == '1' && b == '1')
39         return '1';
40     else
41         return '0';
42 }
43
44 char or(char a ,char b){
45     if(a == '0' && b == '0')
46         return '0';
47     else
48         return '1';
49 }
50
51 void print(char **ans){
52     int i,j=0;
53     int hed[8];
54     char tel[8][10]={"x*y*z*","x*y*z","x*yz*","x*yz","xy*z*","xy*z","yz*","yz"};
55 }
```

```

56     for(i=0 ;i<8 ;i++){
57         if(ans[i][0] == '1'){
58             if(j != 0)
59                 printf("+");
60             printf("%s",tel[i]);
61             j++;
62         }
63     }
64     printf("\n");
65 }
66
67 void f_truth(char *f,char **ans){ //読み取った文字列のx,y,z を 0 と 1 に変換
68     int i,j,k,l,m;
69     int n=0;
70
71     for(i=0 ;i<2 ;i++){
72         for(j=0 ;j<2 ;j++){
73             for(k=0 ;k<2 ;k++){
74                 m=0;
75                 for(l=0 ;l<100 ;l++){
76                     if(f[l] == 0)
77                         break;
78                     switch(f[l]){
79                         case 'x':
80                             if(f[l+1] == '*'){
81                                 ans[n][m]='0'+abs(i-1);
82                                 l++;
83                             }
84                             else
85                                 ans[n][m]='0'+i;
86                             break;
87                         case 'y':
88                             if(f[l+1] == '*'){
89                                 ans[n][m]='0'+abs(j-1);
90                                 l++;
91                             }
92                             else
93                                 ans[n][m]='0'+j;
94                             break;
95                         case 'z':
96                             if(f[l+1] == '*'){
97                                 ans[n][m]='0'+abs(k-1);
98                                 l++;
99                             }
100                            else
101                                ans[n][m]='0'+k;
102                            break;
103                        case '+':
104                            ans[n][m]='+';
105                            break;
106                    }
107                    m++;
108                }
109                n++;
110            }
111        }
112    }

```

```

113 }
114
115 int f_confirm(char **ans){ //ans の中に or がいくつあるか確かめる
116     int i;
117     int sum=0,leg;
118
119     leg=strlen(ans[0]);
120     for(i=0 ;i<100 ;i++){
121         if(ans[0][i] == '+')
122             sum++;
123     }
124     if(leg-sum == sum+1) //or 演算子と数字の数がいい感じだと 0 を返す
125         return 0;
126     else
127         return 1;
128 }
129
130 void f_and(char **ans){ //ans の中の and 演算子だけを計算する
131     int i,j;
132     int n;
133     char temp[N]={}; //ans に直接入れると隙間ができるため、temp を噛ませる
134
135     while(f_confirm(ans) > 0){
136         for(i=0 ;i<8 ;i++){
137             n=0;
138             for(j=0 ;j<N ;j++){
139                 if(ans[i][j] == 0)
140                     break;
141                 switch(ans[i][j]){
142                     case '0':
143                     case '1':
144                         if(ans[i][j+1] == '0' || ans[i][j+1] == '1'){
145                             temp[n]=and(ans[i][j],ans[i][j+1]);
146                             j++;
147                         }
148                     else
149                         temp[n]=ans[i][j];
150                     break;
151                     case '+':
152                         temp[n]=ans[i][j];
153                 }
154                 n++;
155             }
156             strcpy(ans[i],temp);
157         }
158     }
159 }
160
161 void f_or(char **ans){ //or 演算子の計算
162     int i,j;
163     int n;
164     char temp[N]={};
165
166     while(strlen(ans[0]) != 1){
167         for(i=0 ;i<8 ;i++){
168             n=0;
169             for(j=1 ;j<N ;j+=2){ //ans[1] から初めて両はしを計算

```

```

170         if(ans[i][j] == 0)
171             break;
172         temp[n]=or(ans[i][j-1],ans[i][j+1]);
173         n++;
174     }
175     strcpy(ans[i],temp);
176 }
177 }
178 }
179
180
181 int main(void){
182
183     int i,j,k;
184     int n=0;
185     char f[N];
186     char **ans;
187
188     printf("f(x,y,z)= ");scanf("%s",f);
189     usage(f);
190
191     //ans のメモリ確保
192     ans=malloc(sizeof(char *)*8);
193     for(i=0 ;i<8 ;i++)
194         ans[i]=malloc(sizeof(char)*N);
195
196     //ans の初期化
197     for(i=0 ;i<8 ;i++){
198         for(j=0 ;j<N ;j++){
199             ans[i][j]=0;
200         }
201     }
202
203     f_truth(f,ans);
204
205     f_and(ans);
206
207     f_or(ans);
208
209     //表示
210     printf("\n 主加法標準形へ展開すると\n");
211     print(ans);
212
213     free(ans);
214
215     return 0;
216 }

```
