```
/*

        YOGU FINAL


A “snippet” of outline code is given in task2_code.py/ task2_code.c file.   • Teams
Modify the puzzle () function in the file to process both the arrays and sum up the
numbers in D1 which results in a number present in D2.  • Similarly teams complete
all the numbers in D2.  Note: a number in D1 which is used in getting sum of a
number present in D2, cannot be used again. • Print the numbers used in D1for
making the sum in D2.  Note: There can be many possible solutions.




1. take the 3 numbers from the 2nd set

2. subtract each of them with an 1st element of 1st set of numbers

        a. assign to 3 diff variables (x,y,z)

3. Search the rest of the array if we have an element equal to the difference
obtained above

a.      If obtained then the element used to subtract and the obtained are a
possible solution set

        b.      Put those index(s) in another reference array

c.      Add an extra elements say 100 to mark end of soln set

4. calculate length solution set

5.if for a number which has 0 solution ( sum of no TWO numbers of 1st set gives the
sum,implies it needs
  to be sum of more than 2 numbers
)
        a)     The above process as done by dealing with 2 elements will b
implemented for 3to8 elements
                ( as other 2 number will require 2 elements each(total 4))

6)if soln set length is least then it get priority and its values are first
considered

7) used numbers are edited to 100 to mark used
```

```
8)soln set length which is in between is considered

9) last the 1s with large length


*/


#include <stdio.h>
#include <string.h>





int D1[12],D2[4][2];
int i,j,k,i2,j2,k2,x,y,z,xc,yc,zc,at,bt,ct;
int a[20],b[20],c[20];






void construct()
{
        j=0;
        k=0;
        x=0;
        y=0;
        z=0;
        xc=0;
        yc=0;
        zc=0;
        at=0;
        bt=0;
        ct=0;
        for(i=0;i<20;i++)
        {
                a[i]=0;
                b[i]=0;
                c[i]=0;
        }
        i=0;
}
```

```
int siz_arr(int arr[20])
{
    for(i=0;i<20;i++)
       {
           if(arr[i]==100)
           {
               return i;
           }
       }
}




void output(int arr[20],int t1,int t2)
{
    i=0;
    switch(t2)
    {
        case 1:
                printf("%d=%d+%d\n",t1,D1[arr[0]],D1[arr[1]]);
                D1[arr[0]]=100;
                D1[arr[1]]=100;
                break;
        case 2:
            while(1)
            {
                if((D1[arr[i]]!=100)&&(D1[arr[i+1]]!=100))
                {
                    if(t1==(D1[arr[i]]+D1[arr[i+1]]))
                    {
                        printf("%d=%d+%d\n",t1,D1[arr[i]],D1[arr[i+1]]);
                        D1[arr[i]]=100;
                        D1[arr[i+1]]=100;
                        break;
                    }
                    else
                         more_than_2(t1);
                         break;
                }
                i+=2;
            }
            break;
    }
}

void soln_ar(int p,int q ,int r)
```

```
{
    for(i=0;i<12;i++)
    {
        x=p-D1[i];
        y=q-D1[i];
        z=r-D1[i];
        for(j=(i+1);j<12;j++)
        {

                if((x==D1[j]))
                {
                    a[at]=j;
                    a[++at]=i;
                    at++;
                }

                if((y==D1[j]))
                {
                    b[bt]=j;
                    b[++bt]=i;
                    bt++;
                }
                if((z==D1[j]))
                {
                    c[ct]=j;
                    c[++ct]=i;
                    ct++;
                }
        }
    }
}




int more_than_2(int a)
{
    int i,j,k,l,m,n,o,p;
    int x[8]={0,0,0,0,0,0,0,0};
    for(i=0;i<12;i++)
    {
        x[0]=a-D1[i];
        for(j=(i+1);(j<12);j++)
        {

           x[1]=x[0]-D1[j];
           for(k=(j+1);k<12;k++)
           {
               if(D1[k]==x[1])
               {
                    printf("%d=%d+%d+%d\n",a,D1[i],D1[j],D1[k]);
                    D1[i]=100;
                    D1[j]=100;
                    D1[k]=100;
                    return;
               }
               else
               {
                   x[2]=x[1]-D1[k];
                    for(l=(k+1);l<12;l++)
                    {
                        if(D1[l]==x[2])
                        {
                            printf("%d=%d+%d+%d+%d\n",a,D1[i],D1[j],D1[k],D1[l]);
                            D1[i]=100;
                            D1[j]=100;
                            D1[k]=100;
                            D1[l]=100;
                            return;
                        }
                        else
                        {
                            x[3]=x[2]-D1[l];
                            for(m=(l+1);m<12;m++)
                            {
                                if(D1[m]==x[3])
                                {

printf("%d=%d+%d+%d+%d+%d\n",a,D1[i],D1[j],D1[k],D1[l],D1[m]);
                                    D1[i]=100;
                                    D1[j]=100;
                                    D1[k]=100;
                                    D1[l]=100;
                                    D1[m]=100;
                                    return;
                                }
                                else
                                {
                                    x[4]=x[3]-D1[m];
                                    for(n=(m+1);n<12;n++)
                                    {
                                        if(D1[n]==x[4])
                                        {

printf("%d=%d+%d+%d+%d+%d+%d\n",a,D1[i],D1[j],D1[k],D1[l],D1[m],D1[n]);
```

```
                                            D1[i]=100;
                                            D1[j]=100;
                                            D1[k]=100;
                                            D1[l]=100;
                                            D1[m]=100;
                                            D1[n]=100;
                                            return;
                                        }
                                        else
                                        {
                                            x[5]=x[4]-D1[n];
                                            for(o=(n+1);o<12;o++)
                                            {
                                                if(D1[o]==x[5])
                                                {

printf("%d=%d+%d+%d+%d+%d+%d+%d\n",a,D1[i],D1[j],D1[k],D1[l],D1[m],D1[n],D1[o]);
                                                    D1[i]=100;
                                                    D1[j]=100;
                                                    D1[k]=100;
                                                    D1[l]=100;
                                                    D1[m]=100;
                                                    D1[n]=100;
                                                    D1[o]=100;
                                                    return;
                                                }
                                                else
                                                {
                                                    x[6]=x[5]-D1[o];
                                                    for(p=(o+1);p<12;p++)
                                                    {
                                                        if(D1[p]==x[6])
                                                        {

printf("%d=%d+%d+%d+%d+%d+%d+%d+%d\n",a,D1[i],D1[j],D1[k],D1[l],D1[m],D1[n],D1[o],D
1[p]);
                                                            D1[i]=100;
                                                            D1[j]=100;
                                                            D1[k]=100;
                                                            D1[l]=100;
                                                            D1[m]=100;
                                                            D1[n]=100;
                                                            D1[o]=100;
                                                            D1[p]=100;
                                                            return;
                                                        }
                                                    }
                                                }
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
               }
           }
        }
    }
    return 0;
}




void puzzle()
{
    soln_ar(D2[0][1],D2[1][1],D2[2][1]);
    a[at]=100;
    b[bt]=100;
    c[ct]=100;

    xc=siz_arr(a);
    yc=siz_arr(b);
    zc=siz_arr(c);


    if (zc==0)
    {
        more_than_2(D2[2][1]);
    }
    if (yc==0)
    {
        more_than_2(D2[1][1]);
    }
    if (xc==0)
    {
        more_than_2(D2[0][1]);
    }
    if((zc>0)&&((zc<yc)&&(zc<xc)))
    {
        output(c,D2[2][1],1);
    }
    if((yc>0)&&((yc<xc)&&(yc<zc)))
    {
        output(b,D2[1][1],1);
    }
    if((xc>0)&&((xc<yc)&&(xc<zc)))
    {
        output(a,D2[0][1],1);
    }
    if((zc>0)&&((zc>yc)&&(zc<xc))||(zc<yc)&&(zc>xc))
    {
        output(c,D2[2][1],2);
    }
    if((yc>0)&&((yc>xc)&&(yc<zc))||(yc<xc)&&(yc>zc))
    {
        output(b,D2[1][1],2);
    }
    if((xc>0)&&((xc>yc)&&(xc<zc))||(xc<yc)&&(xc>zc))
    {
        output(a,D2[0][1],2);
    }
    if((zc>0)&&((zc>=yc)&&(zc>=xc)))
    {
        output(c,D2[2][1],2);
    }
    if((yc>0)&&((yc>=xc)&&(yc>=zc)))
    {
        output(b,D2[1][1],2);
    }
    if((xc>0)&&((xc>=yc)&&(xc>=zc)))
    {
        output(a,D2[0][1],2);
    }
}




int main()
{
    char str3[20]=".txt";
    for(k2=0;k2<3;k2++)
        {
            char str1[20]="Test_input";
            str1[10]=k2+48;
            strcat(str1,str3);
                printf("%s\n",str1);

            FILE* in_file = fopen( str1, "r"); // read only
            if (! in_file ) // equivalent to saying if ( in_file == NULL )
            {
                printf("\noops, file can't be read\n");
                return 0;
            }

          // attempt to read the next line and store
          // the value in the "number" variable
            for (i2 = 0; i2 < 12; i2++)
            {
                fscanf(in_file, "%d", &D1[i2]);
            }
            j2=0;
            i2=0;
            while ( fscanf(in_file, "%d", & D2[i2][j2] ) == 1 )
            {
                if(j2==1)
                    {
                    i2++;j2=0;
                    }
                    else
                    j2++;
            }
        construct();
        puzzle();
        }
    return 0;
}
```