

```
program coin; {CAN be run on https://www.onlinegdb.com/online\_pascal\_compiler}
```

```
{authors Tom Young, Jeff Greenfield, David Raabe, Joe Culbert}
```

```
const highest_number_of_coins_returned = 100;
```

```
var    coin1, coin2, coin3, coin4 : integer;  
    count1, count2, count3, count4 : integer;  
    cents : integer;
```

```
    i, j, k : integer;
```

```
    num1, num2, num3, num4, total, answer, trial, x : integer;
```

```
    try : array[1..12144] of integer;
```

```
    outfile : text;
```

```
begin
```

```
    assign (outfile, 'info.txt');
```

```
    rewrite (outfile);
```

```
    coin1 := 1; {a 4 coin set always has a penny}
```

```
        for coin2 := coin1 + 1 to 97 do {try all possible 4 coin sets}
```

```
            for coin3 := coin2 + 1 to 98 do
```

```
                for coin4 := coin3 + 1 to 99 do
```

```
                    begin total := 0;
```

```
                        for cents := 1 to 99 do {try each change amount}
```

```
                            begin {calculate each possible combination of coins = cents}
```

```
                                trial := 0;
```

```
                                num1 := cents;
```

```
                                k := num1 div coin4;
```

```
                                for count1 := k downto 0 do
```

```
                                    begin
```

```
                                        num2 := num1 - (count1*coin4);
```

```
                                        j := num2 div coin3;
```

```
                                        for count2 := j downto 0 do
```

```
                                            begin
```

```
                                                num3 := num2 - (count2*coin3);
```

```
                                                i := num3 div coin2;
```

```
                                                for count3 := i downto 0 do
```

```
                                                    begin
```

```
                                                        count4 := num3 - (count3*coin2);
```

```
                                                        trial := 1 + trial;
```

```
                                                        try[trial] := count1+count2+count3+count4;{store}
```

```
                                                    end; end; end;
```

```
                            answer := highest_number_of_coins_returned;
```

```
                    for x := 1 to trial do {find smallest combination}
```

```
                        if try[x] < answer then answer := try[x];
```

```
                        total := total+answer;
```

```
                    end;
```

```
                writeln (outfile, coin1, ', ', coin2, ':', coin3, ':', coin4, ':', total, ':');
```

```
            end;
```

```
        close(outfile);
```

```
    end.
```