```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

void getPaid(float* paid, float* due);
void makeChange(float paid, float due, int* dollar, int* qtr, int* dime,
int* nkl, int* pen);
void printResults(float paid, float due, int dollar, int qtr, int dime,
int nkl, int pen);

int main(void)
{

      //variable declarations

      int dollar = 0,
                   qtr = 0,
                  dime = 0,
                   nkl = 0,
                   pen = 0;

      float paid = 0.0,
                   due = 0.0;

      //functions used
      getPaid(&paid, &due);
      makeChange(paid, due, &dollar, &qtr, &dime, &nkl, &pen);
      printf("\n");
      printResults(paid, due, dollar, qtr, dime, nkl, pen);


      return(0);

}
//gets input from user
void getPaid(float* paid, float* due)
{
      printf("Enter the amount paid.\n");
      scanf("%f", paid);
      printf("Enter the amount due.\n");
      scanf("%f", due);

}

void makeChange(float paid, float due, int *dollar, int *qtr, int *dime,
int *nkl, int *pen)
{


      if (paid > due)
      {
                  *dollar = paid - due; //calcs dollars
                  *dime = (((paid - due) - *dollar) + 0.01) * 10; // calcs
dime using
```

```
                *pen = (((paid - due) - *dollar) * 100.0) - (*dime *
10.0); //calcs pennies

                // the max number of pennies is 4
                //anything above 4 pennies becomes a nickel and a number
of pennies
                if (*pen > 4)
                {
                           *pen = *pen - 5;
                           *nkl = *nkl + 1;
                }

                //max number of dimes is 3
                //anything more becomes a quarter and a number of
nickles/dimes
                if (*dime >= 3)
                {
                           *dime = *dime * 10;

                           while (*dime > 25)
                           {
                                 *dime = *dime - 25;
                                 *qtr = *qtr + 1;
                           }

                           if (*dime == 5 || *dime == 15)
                           {
                                 *nkl = *nkl + 1;
                           }

                           *dime = *dime / 10;
                }
     }
}



//simply print the results
void printResults(float paid, float due, int dollar, int qtr, int dime,
int nkl, int pen)
{


     printf("Change due:\n");
     printf("Dollars:%d\n", dollar);
     printf("Dimes:%d\n", dime);
     printf("Pennies:%d\n", pen);
     printf("Quarters:%d\n", qtr);
     printf("Nickels:%d\n", nkl);
}



//FIRST ATTEMPT AT CALCULATING CHANGE DUE
```

```
/*

      if (paid > due)
      {
                  *dollar = paid - due;
                  *dime = (((paid - due) - *dollar)) * 10;
                  *pen = (((paid - due) - *dollar) * 100.0) - (*dime *
10.0);

                  if (*pen > 4)
                  {
                              *pen = *pen - 5;
                              *nkl = *nkl + 1;
                  }

                  if (*dime >= 3)
                  {
                              *dime = *dime * 10;

                              while (*dime > 25)
                              {
                                    *dime = *dime - 25;
                                    *qtr = *qtr + 1;
                              }

                              if (*dime == 5 || *dime == 15)
                              {
                                    *nkl = *nkl + 1;
                              }

                              *dime = *dime / 10;
                  }


      }*/

//SECOUND ATTEMPTED AT CALCULATING CHANGE

/*
float change = 0.0;

      change = paid - due;
      *dollar = change;
      *dime = (change - *dollar) * 10;

      while (*dime >= 3)
      {
                  *dime = *dime - 1;
                  change = change - 0.10;
      }

      *pen = ((change - *dollar) * 10) / 100;
```

```
while (*pen > 4)
{
            *pen = *pen - 1;
            change = change - 0.01;
}

*qtr = ((change - *dollar) * 10) / 25;

while (*qtr >= 3)
{
            *qtr = *qtr - 1;
            change = change - 0.25;
}

*nkl = ((change - *dollar) * 10) / 5;

while (*nkl >= 1)
{
            *qtr = *qtr - 1;
            change = change - 0.05;
}

*/
```