CS100 Tutorial 4 (Fall semester, 2018)

Solutions

Q1.

Suggested code:

```
int intersect(struct circle c1, struct circle c2)
{
   double a, b;

   a = c1.x - c2.x;
   b = c1.y - c2.y;
   return (sqrt(a * a + b * b) <= (c1.radius + c2.radius));
}

int contain(struct circle *c1, struct circle *c2)
{
   double a, b;

   a = c1->x - c2->x;
   b = c1->y - c2->y;
   return (c1->radius  >= (c2->radius  + sqrt(a * a + b * b) ));
}
```

Q2.

Suggested code:

```
void getInput(leaveRecord list[], int * n)
{
   *n = 0;
   while (scanf("%d %d %d", &list[*n].id, &list[*n].totalLeave,
         &list[*n].leaveTaken) != EOF) {
      (*n)++;
   }
}

int mayTakeLeave(leaveRecord list[], int id, int leave, int n)
{
   int p;

   for (p = 0; p < n; p++)
      if (list[p].id == id)
```

```
         return (list[p].totalLeave >=
                list[p].leaveTaken + leave);

   return -1;
}
```

Q3.

Suggested code:

```
int sumUp(int n)
{
   if (n == 1)
      return 1;
   else
      return n + sumUp(n-1);
}


int sumUp2(int n)
{
   int i, sum;
   sum = 0;
   for (i=1; i <= n; i++) {
      sum += i;
   }
   return sum;
}
```

Q4.

Suggested code:

```
float power1(float x, int n)
{
     if (n == 0)
          return 1;
     else if (n < 0)
          return power1(x, n+1) / x;
     else
          return power1(x, n-1) * x;
}

void power2(float x, int n, float *y)
{
     if (n == 0) {
          *y = 1;
     } else if (n < 0) {
          power2(x, n+1, y);
          *y /= x;
     } else {
          power2(x, n-1, y);
          *y *= x;
     }
}
```