```c
// Program 1
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int i;
    for(i = 0; i < 10; ++i)
    {
        printf("i = %d\n", i);
        if(i % 2 == 0)
            continue;
        printf("i^2 = %d\n", i * i);
    }
    exit(0);
}

// Program 2
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int i;
    for(i = -16; i < 16; ++i)
    {
        if(i % 4 == 0)
            continue;
        printf("i = %d\n", i);
    }
    exit(0);
}

// Program 3
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int i;
    i = 0;
    while(i < 10)
    {
        if(i % 2 == 0)
        {
            continue;
        }
        printf("i = %d\n", i);
        i += 1;
    }

    exit(0);
}

// Program 4
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int i;
    i = 0;
    while(i < 10)
    {
        if(i%2 == 0)
        {
            i += 1;
            continue;
        }
```

```
70          printf("i = %d\n", i);
71          i += 1;
72      }
73      exit(0);
74  }
75
76  // Program 5
77  #include <stdio.h>
78  #include <stdlib.h>
79
80  int main(void)
81  {
82      int i;
83      for(i = 0; i < 8; ++i)
84      {
85          if(i % 2 == 0)
86              continue;
87          else if(i % 4 == 0)
88              break;
89          else
90              printf("i = %d\n", i);
91      }
92      exit(0);
93  }
94
95  // Program 6
96  #include <stdio.h>
97  #include <stdlib.h>
98
99  int main(void)
100 {
101     int i;
102     for(i = 0; i < 20; ++i)
103     {
104         if((i * i) > 50)
105             break;
106         else
107             printf("i = %d\n", i);
108     }
109     exit(0);
110 }
111
112 // Program 7
113 #include <stdio.h>
114 #include <stdlib.h>
115
116 int main(void)
117 {
118     int i;
119     int k;
120     int flag;
121     for(i = 1; i < 20; ++i)
122     {
123         if(i == 1 && i == 2)
124             continue;
125         flag = 0;
126         for(k = 2; k < i; ++k)
127         {
128             if(i % k == 0)
129             {
130                 flag = 1;
131                 break;      // If break or continue appear in nested loop then
132                             // their effect is applicable only to the inner most loop
133                             // of which they are part.
134                             // e.g. This break will break out of inner loop for(k=2; k
                            < i; ++k)
135                             // but will not break from the outer loop for(i = 0; i <
                            20; ++i)
136                             // Same comments can be extended for the continue statement
```

```
            }
        }

        if(flag == 1)
            printf("i = %d\n", i) ;
        else
            continue;
    }
    exit(0);
}

// Program 8
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int i, j;
    for(i = 0; i < 5; ++i)
        for(j = 0; j < 5; ++i)
            if(i > j)
                continue;
            else
                printf("i=%d is less than or equal to j=%d\n", i, j);
    exit(0);
}

// Program 9
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int i, j;
    for(i = 0; i < 10; ++i)
    {
        if(i % 2 == 1)
            continue;
        for(j = 0; j < 10; ++i)
        {
            if(j % 2 == 0)
                continue;
            printf("i=%d, j=%d\n", i, j);
        }
    }
    exit(0);
}

// Program 10
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int i, j, k;
    int flag1, flag2;

    for(i = 1; i <= 10; ++i)
    {
        flag1 = 1;
        for(k = 2; k < i; ++k)
            if(i % k == 0)
                flag1 = 0;

        if(flag1 == 1)
        {
            for(j = 1; j <= 10; ++j)
            {
                flag2 = 1;
```

```
206                     for(k=2; k < j; ++j)
207                         if(j % k == 0)
208                         {
209                             flag2 = 0;
210                             break;
211                         }
212                     if(flag2 == 0)
213                         printf("Prime(i=%d), Composite(j=%d)\n", i, j);
214                 }
215             }
216         }
217         exit(0);
218     }
```