

## TUGAS!

- **Perulangan**

1. Buatlah algoritma yang membaca sembarang karakter dan mencetaknya ke layar. Buatlah algoritma untuk dua kasus:

- (a) Jumlah karakter yang dibaca diketahui, yaitu N buah (baca N terlebih dahulu).
- (b) Jumlah karakter yang dibaca tidak diketahui, tetapi proses pembacaan berakhir jika karakter yang dimasukkan adalah karakter titik (karakter titik tidak ikut dicetak).

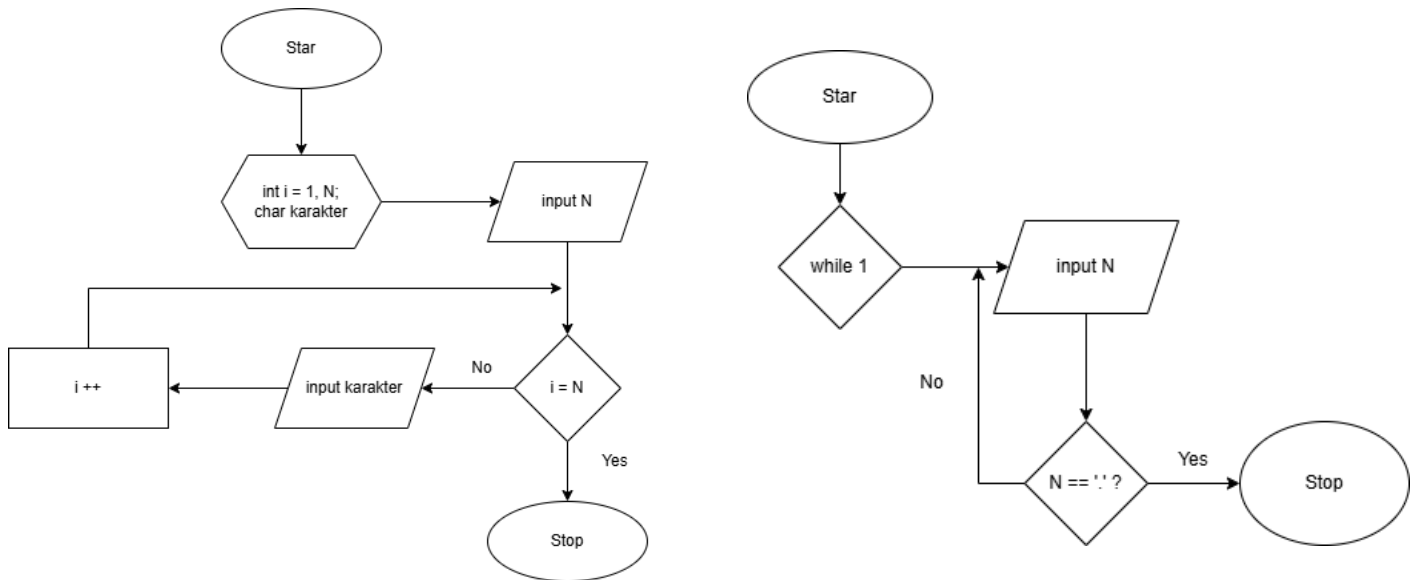
**Jawab:**

a. *Source Code*

```
1  #include <stdio.h>
2
3  int main() {
4      int N, i;
5      char karakter;
6
7      printf("Masukkan jumlah karakter: ");
8      scanf("%d", &N);
9
10     printf("Masukkan %d karakter:\n", N);
11
12     for (i = 0; i < N; i++) {
13         scanf(" %c", &karakter);
14         printf("%c", karakter);
15     }
16
17     printf("\n");
18
19     return 0;}
```

```
1  #include <stdio.h>
2
3  int main(){
4      char N;
5      printf("Masukkan karakter: ");
6      while (1)
7      {
8          scanf("%c", &N);
9          if (N=='.'){
10             break;
11         }
12         printf ("%c", N);
13     }
14
15     return 0;
16 }
```

*b. Flowchart*



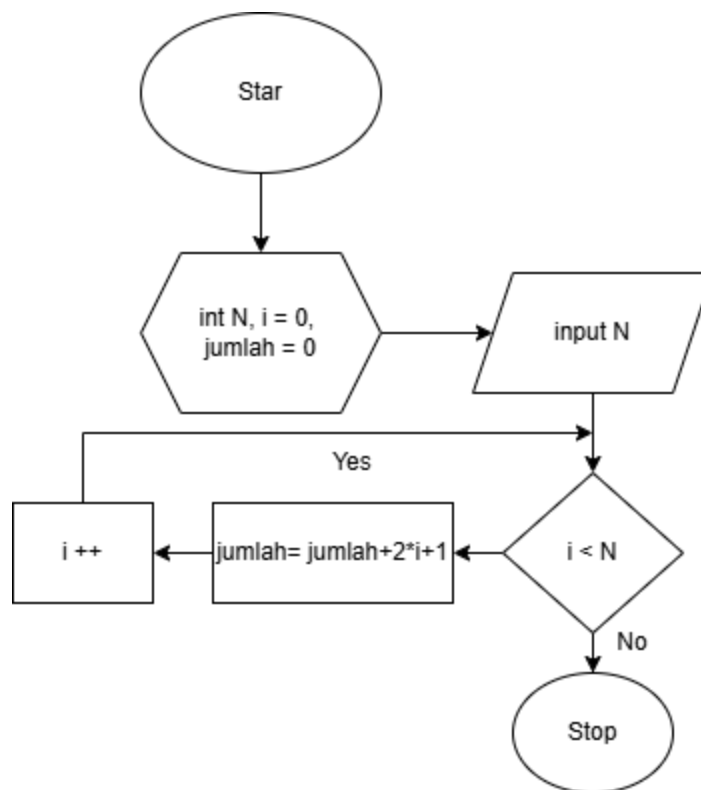
2. Buatlah algoritma untuk menghitung jumlah N buah bilangan ganjil pertama (yaitu, 1 + 3 + 5 + ...). Catatan: N adalah bilangan bulat tidak negatif.

**Jawab:**

*a. Source Code*

```
1  #include<stdio.h>
2
3  int main(){
4      int i, N, jumlah= 0;
5
6      printf("Masukkan nilai N: ");
7      scanf ("%d", &N);
8
9      for (i=0; i<N; i++){
10         jumlah += 2*i+1;
11     }
12     printf ("jumlah %d bilangan ganjil pertama ialah: %d", N, jumlah);
13     return 0;}
```

*b. Flowchart*



3. Buatlah algoritma untuk menghitung jumlah bilangan ganjil dari 1 sampai N (yaitu,  $1 + 3 + 5 + \dots + N$ ). (Perhatikan perbedaan soal latihan ini dengan soal nomor 1.)

**Jawab:**

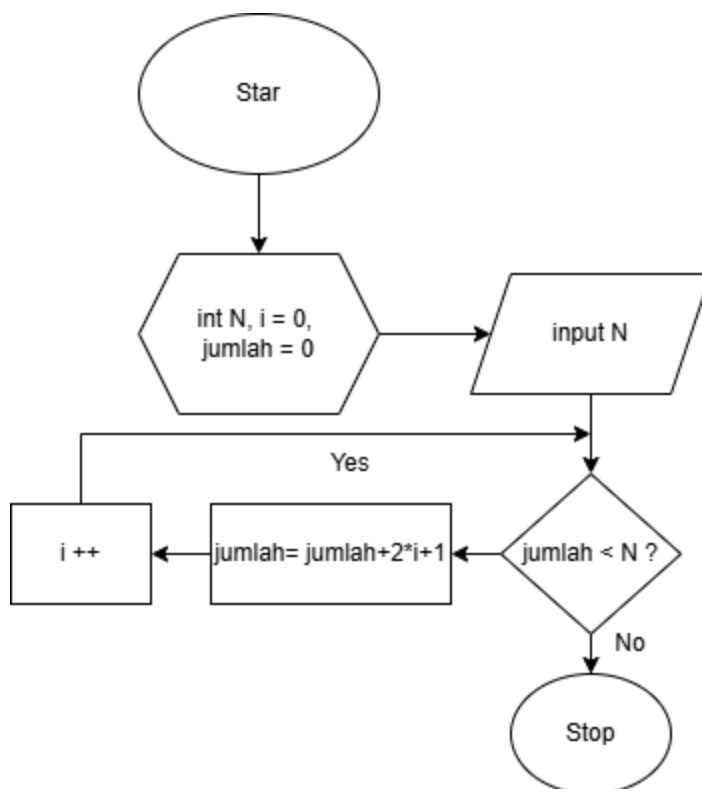
*a. Source Code*

```

1  #include<stdio.h>
2
3  int main(){
4      int i, N, jumlah= 0;
5
6      printf("Masukkan nilai N: ");
7      scanf ("%d", &N);
8
9      for (i=0; jumlah < N; i++){
10         jumlah += 2*i+1;
11     }
12     printf ("jumlah bilangan ganjil pertama dari 1 sampai %d  adalah %d", N, jumlah);
13     return 0;
14 }

```

*b. Flowchart*



4. Tuliskan program C untuk mencetak gambar segitiga bintang sebanyak N baris, setiap baris ke-i berisi i buah bintang. Nilai N dibaca dari papan kunci.

Contoh: masukan: N = 5

Keluaran:

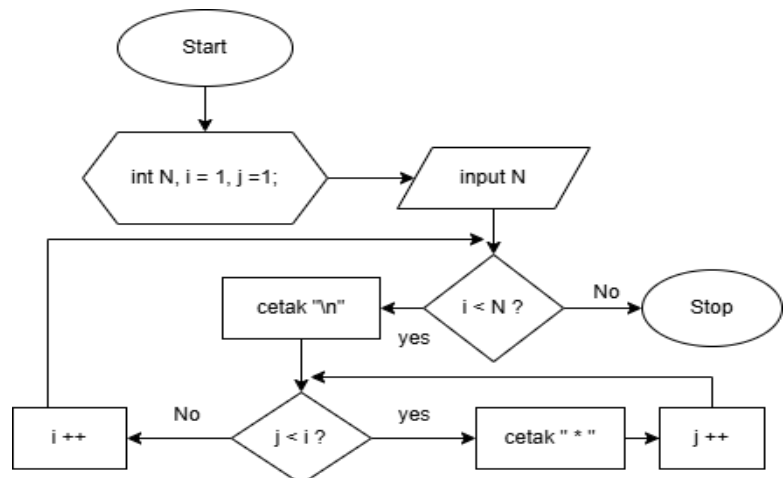
```
*
**
***
****
*****
```

**Jawab:**

a. Source Code

```
1  #include <stdio.h>
2
3  int main() {
4      int i, N;
5      printf ("Masukkan nilai N: ");
6      scanf ("%d", &N);
7
8      for (i=1; i<=N; i++){
9          printf ("\n");
10         for (int j=1; j<=i; j++){
11             printf ("*");
12         }
13     }
14     return 0;
15 }
```

b. Flowchart

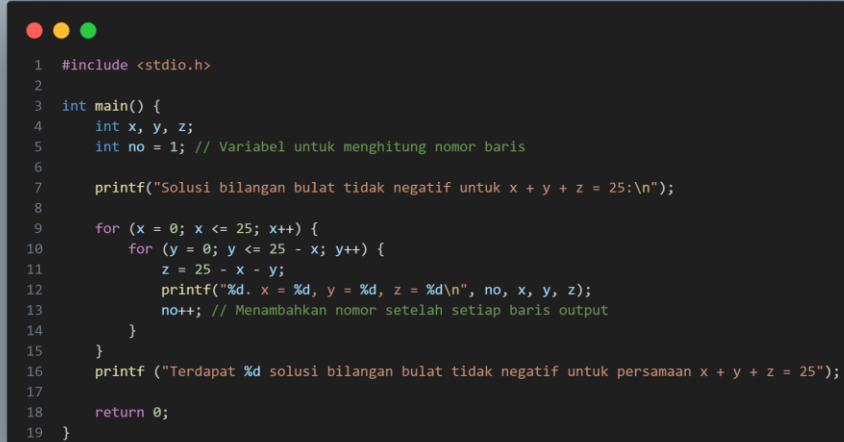


5. Tuliskan algoritma untuk menampilkan semua solusi bilangan bulat tidak negatif dari persamaan berikut:

$$x + y + z = 25$$

Jawab:

a. Source Code



```
1  #include <stdio.h>
2
3  int main() {
4      int x, y, z;
5      int no = 1; // Variabel untuk menghitung nomor baris
6
7      printf("Solusi bilangan bulat tidak negatif untuk x + y + z = 25:\n");
8
9      for (x = 0; x <= 25; x++) {
10         for (y = 0; y <= 25 - x; y++) {
11             z = 25 - x - y;
12             printf("%d. x = %d, y = %d, z = %d\n", no, x, y, z);
13             no++; // Menambahkan nomor setelah setiap baris output
14         }
15     }
16     printf ("Terdapat %d solusi bilangan bulat tidak negatif untuk persamaan x + y + z = 25");
17
18     return 0;
19 }
```

b. Pseudocode

**Algoritma:** Solusi Bilangan Tidak Negatif

**Definisi Variabel:**

integer x, y, z, no

**Rincian Langkah:**

1. Cetak "Solusi bilangan bulat tidak negatif untuk x + y + z = 25:"
2. Set no ke 1
3. **For** x dari 0 hingga 25 **do**

**For** y dari 0 hingga (25 - x) **do**

Set z = 25 - x - y

Cetak no, "x = ", x, " y = ", y, " z = ", z

Set no = no + 1

**EndFor**

**EndFor**

4. Cetak "Terdapat", no , "solusi bilangan bulat tidak negatif untuk persamaan  $x + y + z = 25$ "

6. Buatlah algoritma yang akan mengonversi bilangan bulat positif ke angka romawinya. Rancanglah algoritma tersebut sehingga pengonversian tersebut dapat dilakukan berulang kali sampai nilai nol dibaca dari piranti masukan.

**Jawab:**

a. Source Code

```
1  #include <stdio.h>
2  #include <string.h>
3
4  void konversiKeRomawi(int nilai, char hasil[]) {
5      int nilaiArab[] = {1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1};
6      char *angkaRomawi[] = {"M", "CM", "D", "CD", "C", "XC", "L", "XL", "X", "IX", "V", "IV", "I"};
7
8      hasil[0] = '\0';
9
10     for (int i = 0; i < 13; i++) {
11         while (nilai >= nilaiArab[i]) {
12             strcat(hasil, angkaRomawi[i]);
13             nilai -= nilaiArab[i];
14         }
15     }
16 }
17
18 int main() {
19     int angka;
20     char romawi[100];
21
22     while (1) {
23         printf("Masukkan bilangan bulat positif (0 untuk keluar): ");
24         scanf("%d", &angka);
25
26         if (angka == 0) {
27             printf("Program selesai.\n");
28             break;
29         } else if (angka < 0) {
30             printf("Input harus bilangan positif!\n");
31             continue;
32         }
33
34         konversiKeRomawi(angka, romawi);
35         printf("Angka Romawi: %s\n", romawi);
36     }
37
38     return 0;
39 }
40
```

b. Pseudocode

### ***Fungsi Konversi ke Romawi:***

**Algoritma :** Konversi ke Romawi

**Definisi Variabel:**

- integer nilai
- array integer nilaiArab dengan elemen {1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1}
- array string angkaRomawi dengan elemen {"M", "CM", "D", "CD", "C", "XC", "L", "XL", "X", "IX", "V", "IV", "I"}
- string hasil

**Rincian Langkah:**

1. Set hasil ke string kosong
2. For i dari 0 hingga 12 do
  - While nilai  $\geq$  nilaiArab[i] do
    - Gabungkan hasil dengan angkaRomawi[i]
    - Set nilai = nilai - nilaiArab[i]
  - EndWhile
3. EndFor

### ***Algoritma Utama Konversi ke Romawi:***

**Algoritma Utama:** Konversi ke Romawi

**Definisi Variabel:**

integer angka

string romawi

**Rincian Langkah:**

- **Loop** selama **do**
  - a. Cetak "Masukkan bilangan bulat positif (0 untuk keluar): "
  - b. Baca angka
  - c. **If** angka = 0 **then**
    - Cetak "Program selesai."
    - Break loop
  - d. **Else if** angka < 0 **then**
    - Cetak "Input harus bilangan positif!"
    - Continue loop
  - e. **Else**



Panggil prosedur Konversi ke Romawi dengan parameter angka dan romawi

Cetak "Angka Romawi: ", romawi

f. **EndIf**

▪ **EndLoop**

7. Seseorang mempunyai tabungan di sebuah bank. Ia dapat menyetor dan mengambil uangnya di bank tersebut, namun jumlah saldo minimum yang harus disisakan di dalam adalah Rp10.000. Ini artinya, jika saldonya Rp10.000, ia tidak dapat mengambil uang lagi. Kode transaksi untuk menyetor adalah 0 dan kode transaksi untuk mengambil adalah 1. Buatlah algoritma yang menyimulasikan transaksi yang dilakukan orang tersebut. Algoritma menerima masukan berupa kode transaksi dan jumlah uang yang disetor/diambil. Rancanglah algoritma tersebut sehingga memungkinkan penabung dapat melakukan transaksi berulang kali sampai saldo yang tersisa Rp10.000 atau jumlah uang yang diambil lebih besar dari saldonya.

**Catatan:** nilai uang yang diambil selalu merupakan bilangan bulat.

**Jawab:**

a. *Source Code*

```
1 #include <stdio.h>
2
3 int main() {
4     int saldo, kode_transaksi, jumlah_uang;
5
6     printf("Masukkan saldo awal: ");
7     scanf("%d", &saldo);
8
9     while (1) {
10        printf("Masukkan kode transaksi (0 untuk menyetor, 1 untuk mengambil): ");
11        scanf("%d", &kode_transaksi);
12
13        if (kode_transaksi != 0 && kode_transaksi != 1) {
14            printf("Kode transaksi tidak valid. Silakan masukkan 0 atau 1.\n");
15            continue;
16        }
17
18        printf("Masukkan jumlah uang: ");
19        scanf("%d", &jumlah_uang);
20
21        if (kode_transaksi == 0) {
22
23            saldo += jumlah_uang;
24            printf("Berhasil menyetor %d. Saldo Anda sekarang: %d\n", jumlah_uang, saldo);
25        } else if (kode_transaksi == 1) {
26
27            if (saldo - jumlah_uang < 10000) {
28                printf("Tidak dapat mengambil uang. Saldo minimum harus Rp10.000.\n");
29            } else {
30                saldo -= jumlah_uang;
31                printf("Berhasil mengambil %d. Saldo Anda sekarang: %d\n", jumlah_uang, saldo);
32            }
33
34            if (saldo <= 10000) {
35                printf("Saldo Anda sudah mencapai batas minimum. Transaksi dihentikan.\n");
36                break;
37            }
38        }
39
40        return 0;
41    }
42
43 }
```

*b. Pseudocode*

**Algoritma:** Transaksi Bank Sederhana

**Definisi Variabel:**

- integer saldo, kode\_transaksi, jumlah\_uang

**Rincian Langkah:**

1. Cetak "Masukkan saldo awal: "
2. Baca saldo
3. **Loop** selama do
  - Cetak "Masukkan kode transaksi (0 untuk menyetor, 1 untuk mengambil): "
  - Baca kode\_transaksi
  - **If** (kode\_transaksi  $\neq$  0 dan kode\_transaksi  $\neq$  1) **then**
    - Cetak "Kode transaksi tidak valid. Silakan masukkan 0 atau 1."
    - Continue loop
  - **EndIf**
  - Cetak "Masukkan jumlah uang: "
  - Baca jumlah\_uang
  - **If** kode\_transaksi = 0 **then**
    - Set saldo = saldo + jumlah\_uang
    - Cetak "Berhasil menyetor ", jumlah\_uang, ". Saldo Anda sekarang: ", saldo
  - **Else if** kode\_transaksi = 1 **then**
    - **If** (saldo - jumlah\_uang) < 10000 **then**
      - Cetak "Tidak dapat mengambil uang. Saldo minimum harus Rp10.000."
    - **Else**
      - Set saldo = saldo - jumlah\_uang
      - Cetak "Berhasil mengambil ", jumlah\_uang, ". Saldo Anda sekarang: ", saldo
    - **EndIf**
  - **EndIf**
  - **If** saldo  $\leq$  10000 **then**
    - Cetak "Saldo Anda sudah mencapai batas minimum. Transaksi dihentikan."
    - Break loop
  - **EndIf**
4. **EndLoop**

- **Pemilihan**

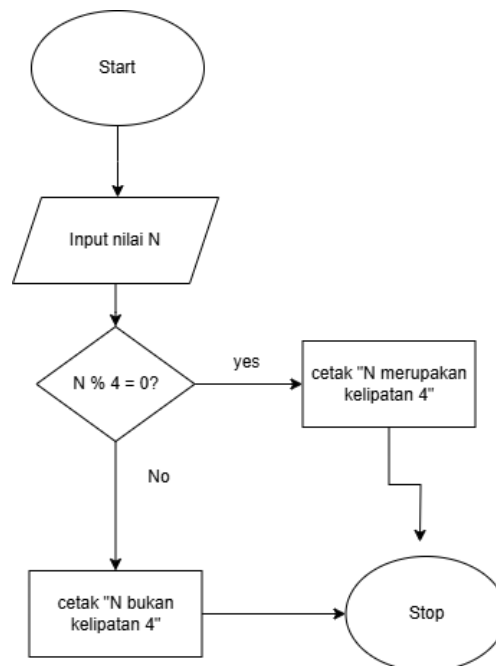
1. Buatlah algoritma yang membaca sebuah bilangan bulat positif lalu menentukan apakah bilangan tersebut merupakan kelipatan 4.

**Jawab:**

*a. Source Code*

```
1  #include <stdio.h>
2
3  int main() {
4      int bilangan;
5
6      printf ("Masukkan bilangan yang ingin di cek: "); scanf ("%d", &bilangan);
7
8      if (bilangan % 4 == 0){
9          printf ("Bilangan %d merupakan kelipatan 4\n", bilangan);
10     } else {
11         printf("Bilangan %d bukan kelipatan 4\n", bilangan);
12     }
13
14     return 0;
15 }
```

*b. Flowchart*



2. Pasar swalayan X memberikan diskon harga bagi pembeli yang nilai total belanjanya lebih dari Rp100.000. Tulislah algoritma untuk menentukan harga belanja setelah

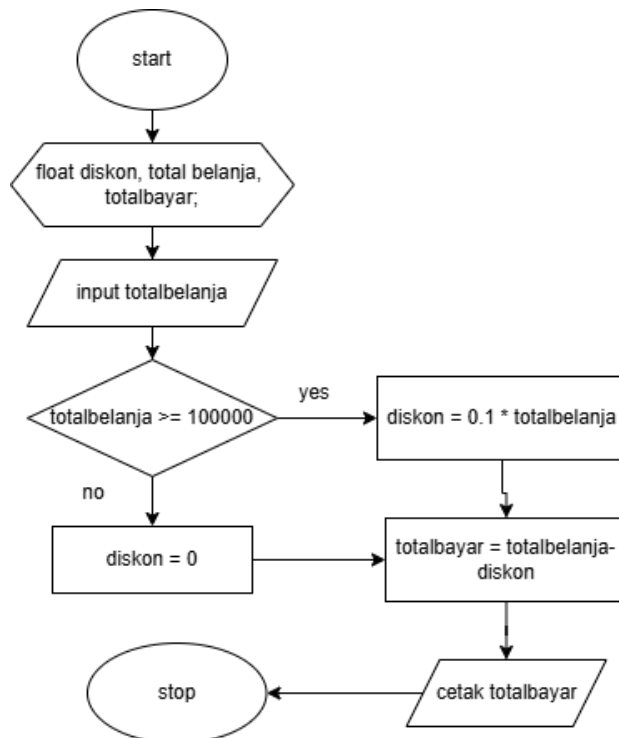
dikurangi diskon. Data masukan adalah nilai total belanja pembeli, sedangkan keluarannya adalah diskon harga dan nilai belanja setelah dikurangi diskon.

**Jawab:**

a. Source Code

```
1 #include <stdio.h>
2
3 int main (){
4     float diskon, totalbelanja, totalbayar;
5     printf ("Masukkan total belanja: ");
6     scanf ("%f", &totalbelanja);
7
8     if (totalbelanja > 100000){
9         diskon = 0.1 * totalbelanja;
10    }else {
11        diskon = 0;
12    }
13    totalbayar = totalbelanja - diskon;
14    printf ("Total harga yang dibayar adalah sebesar Rp.%.2f dengan total diskon yang didapat sebesar Rp.%.2f\n", totalbayar, diskon);
15
16    return 0;
17 }
```

b. Flowchart



3. Tulislah algoritma yang membaca tiga buah bilangan bulat, lalu mengurutkan tiga buah bilangan tersebut dari nilai yang kecil ke nilai yang besar. Keluaran adalah tiga buah bilangan yang terurut.

**Jawab:**

*a. Source Code*

```
1  #include<stdio.h>
2
3  int main (){
4      int bil1, bil2, bil3;
5      int temp;
6
7      printf("Masukkan bilangan pertama: ");
8      scanf("%d", &bil1);
9      printf("Masukkan bilangan kedua: ");
10     scanf("%d", &bil2);
11     printf("Masukkan bilangan ketiga: ");
12     scanf("%d", &bil3);
13
14     if (bil2 < bil1 && bil2 < bil3){
15         temp = bil1;
16         bil1 = bil2;
17         bil2 = temp;
18     }else if (bil3 < bil1 && bil3 < bil2){
19         temp = bil1;
20         bil1 = bil3;
21         bil3 = temp;
22     }
23
24     if (bil3 < bil2){
25         temp = bil2;
26         bil2 = bil3;
27         bil3 = temp;
28     }
29
30     printf ("Urutan bilangan dari yang terkecil adalah:\n 1. %d\n 2. %d\n 3. %d\n", bil1, bil2, bil3);
31     return 0;
32 }
```

*b. Pseudocode*

**Algoritma:** Urutkan 3 Bilangan

**Definisi Variabel:**

- integer bil1, bil2, bil3, temp

**Rincian Langkah:**

1. Cetak "Masukkan bilangan pertama: "
2. Baca bil1
3. Cetak "Masukkan bilangan kedua: "
4. Baca bil2
5. Cetak "Masukkan bilangan ketiga: "

6. Baca bil3
7. **If** (bil2 < bil1) dan (bil2 < bil3) **then**
  - o Set temp = bil1
  - o Set bil1 = bil2
  - o Set bil2 = temp
8. **Else if** (bil3 < bil1) dan (bil3 < bil2) **then**
  - o Set temp = bil1
  - o Set bil1 = bil3
  - o Set bil3 = temp
9. **EndIf**
10. **If** bil3 < bil2 **then**
  - o Set temp = bil2
  - o Set bil2 = bil3
  - o Set bil3 = temp
11. **EndIf**
12. Cetak "Urutan bilangan dari yang terkecil adalah:"
13. Cetak "1. ", bil1
14. Cetak "2. ", bil2
15. Cetak "3. ", bil3
- 16.

4. Tulislah algoritma yang membaca panjang (integer) tiga buah sisi sebuah segitiga, a, b, dan c, yang dalam hal ini  $a \leq b \leq c$ , lalu menentukan apakah ketiga sisi tersebut membentuk segitiga siku-siku, segitiga lancip, atau segitiga tumpul (Petunjuk: gunakan hukum Pythagoras).

**Jawab:**

a. *Source Code*

```

1  #include<stdio.h>
2
3  int main() {
4      int a, b, c;
5      printf (" Masukkan sisi-sisi yang akan membentuk segitiga dimana sisi a < b < c:\n");
6      printf (" Masukkan sisi a: "); scanf ("%d", &a);
7      printf (" Masukkan sisi b: "); scanf ("%d", &b);
8      printf (" Masukkan sisi c: "); scanf ("%d", &c);
9
10     if (a+b < c){
11         printf("Bukan sisi segitiga\n");
12     } else {
13         if (a*a + b*b == c*c){
14             printf ("Merupakan segitiga siku-siku\n");
15         } else if (a*a + b*b < c*c){
16             printf ("Merupakan segitiga lancip\n");
17         } else if (a*a + b*b > c*c){
18             printf ("Merupakan segitiga tumpul\n");
19         }
20     }}

```

*b. Flowchart*

**Algoritma:** Menentukan Jenis Segitiga

**Definisi Variabel:**

- integer a, b, c

**Rincian Langkah:**

1. Cetak "Masukkan sisi-sisi yang akan membentuk segitiga (dengan asumsi sisi  $a < b < c$ ):"
2. Cetak "Masukkan sisi a: "
3. Baca a
4. Cetak "Masukkan sisi b: "
5. Baca b
6. Cetak "Masukkan sisi c: "
7. Baca c
8. **If**  $(a + b) < c$  **then**
  - Cetak "Bukan sisi segitiga"
9. **Else**
  - **If**  $(a^2 + b^2) = c^2$  **then**
    - Cetak "Merupakan segitiga siku-siku"
  - **Else if**  $(a^2 + b^2) < c^2$  **then**
    - Cetak "Merupakan segitiga lancip"
  - **Else if**  $(a^2 + b^2) > c^2$  **then**
    - Cetak "Merupakan segitiga tumpul"
  - **EndIf**
10. **EndIf**

5. Tulislah algoritma yang membaca sebuah karakter digit ('0'..'9') lalu mengonversinya menjadi nilai integer (0..9). Misalnya, jika dibaca karakter '5', maka nilai konversinya ke integer adalah 5.

Buatlah algoritma untuk dua keadaan berikut:

- (a) Karakter digit yang dibaca diasumsikan sudah benar terletak dalam rentang '0'..'9'.
- (b) Karakter yang dibaca mungkin bukan digit '0'..'9'. Jika karakter yang dibaca bukan karakter digit, maka hasil konversinya diasumsikan bernilai -99.

**Jawab:**

*a. Source Code*

```
1 #include <stdio.h>
2
3 int main() {
4     char angka;
5     char *angka2;
6     int opsi, nilai;
7
8     printf("OPSI:\n\t 1. Konversi karakter digit rentang 0-9 ke integer\n\t 2. Konversi karakter digit yang bukan rentang 0-9 ke integer\n\n");
9
10    while (1){
11        printf ("Masukkan opsi yang diinginkan [1\\2]: ");
12        scanf ("%d", &opsi);
13
14        if (opsi == 1){
15            printf ("Masukkan angka yang diinginkan: ");
16            scanf ("%c", &angka);
17            nilai = angka - '0';
18            printf ("Hasil angka yang telah dikonversi ialah integer %d", nilai);
19            break;
20        } else if (opsi == 2){
21            printf ("Masukkan angka yang diinginkan: ");
22            scanf ("%s", &angka2);
23            nilai = -99;
24            printf ("Hasil angka yang telah dikonversi ialah integer %d", nilai);
25            break;
26        } else {
27            printf ("Ops yang dimasukkan tidak valid\n\n");
28        }
29    }
30    return 0;}
```

*b. Flowchart*

**Algoritma :** Konversi Karakter Digit ke Integer

**Definisi Variabel:**

- char angka, pointer angka2
- integer opsi, nilai

**Rincian Langkah:**

1. Cetak "OPSI:"
2. Cetak "\t1. Konversi karakter digit rentang 0-9 ke integer"
3. Cetak "\t2. Konversi karakter digit yang bukan rentang 0-9 ke integer"



#### 4. **Loop selama do**

- Cetak "Masukkan opsi yang diinginkan [1\2]: "
- Baca opsi
- **If** opsi = 1 **then**
  - Cetak "Masukkan angka yang diinginkan: "
  - Baca angka
  - Set nilai = angka - '0'
  - Cetak "Hasil angka yang telah dikonversi ialah integer ",  
nilai
  - Break loop
- **Else if** opsi = 2 **then**
  - Cetak "Masukkan angka yang diinginkan: "
  - Baca angka2
  - Set nilai = -99
  - Cetak "Hasil angka yang telah dikonversi ialah integer ",  
nilai
  - Break loop
- **Else**
  - Cetak "Opsi yang dimasukkan tidak valid"
- **EndIf**

#### 5. **EndLoop**

6. Jika kita berbelanja di pasar swalayan/supermarket, nilai total belanja kita seringkali bukan kelipatan pecahan rupiah yang berlaku. Misalnya, nilai total belanja adalah Rp19.212,-. Andaikan saat ini pecahan rupiah yang berlaku paling kecil Rp25,-. Selain itu, juga ada pecahan Rp50,- dan Rp100,-. Umumnya kasir pasar swalayan membulatkan nilai belanja ke pecahan yang terbesar. Jadi Rp19.212, dibulatkan menjadi Rp19.225,-. Hal ini jelas merugikan konsumen. Misalkan Anda memiliki pasar swalayan yang jujur dan tidak merugikan pembeli, sehingga jika ada nilai belanja yang bukan kelipatan pecahan yang ada, maka nilai belanja itu dibulatkan ke pecahan terendah. Jadi, Rp19.212,- dibulatkan menjadi Rp19.200. Tulislah algoritma yang membaca nilai belanja (integer) lalu membulatkannya ke nilai uang dengan pecahan terendah.

**Jawab:**

a. Source Code

```
1  #include <stdio.h>
2
3  int main(){
4      int totalbelanja, totalbayar;
5
6      printf("Masukkan total belanja: ");
7      scanf("%d", &totalbelanja);
8      totalbayar = (totalbelanja/25) * 25;
9
10     printf ("Total harga yang dibayar setelah dibulatkan dari Rp.%d menjadi Rp.%d\n", totalbelanja, totalbayar);
11     return 0;
12 }
```

b. Pseudocode

**Algoritma :** Pembulatan Total Harga Belanja

**Definisi Variabel:**

- integer totalbelanja
- integer totalbayar

**Rincian Langkah:**

1. Cetak "Masukkan total belanja: "
2. Baca totalbelanja
3. Set  $totalbayar = (totalbelanja / 25) * 25$
4. Cetak "Total harga yang dibayar setelah dibulatkan dari Rp.", totalbelanja, " menjadi Rp.", totalbayar
5. Selesai

7. (a) Tuliskan algoritma yang membaca bilangan bulat positif dalam rentang 1 sampai 10, lalu mengonversinya ke dalam angka romawi.
- (b) Kembangkan algoritma (a) di atas sehingga dapat mengonversi bilangan bulat positif sembarang ke dalam angka romawinya.

Jawab:

a. Source Code

```
1  #include <stdio.h>
2
3  int main() {
4      int angka;
5
6      printf("Masukkan bilangan bulat (1 - 10): ");
7      scanf("%d", &angka);
8
9      printf("Angka Romawi: ");
10     switch(angka) {
11         case 1: printf("I"); break;
12         case 2: printf("II"); break;
13         case 3: printf("III"); break;
14         case 4: printf("IV"); break;
15         case 5: printf("V"); break;
16         case 6: printf("VI"); break;
17         case 7: printf("VII"); break;
18         case 8: printf("VIII"); break;
19         case 9: printf("IX"); break;
20         case 10: printf("X"); break;
21         default: printf("Input tidak valid!"); break;}
22
23     printf("\n");
24
25     return 0;}
```

```
1  #include <stdio.h>
2  #include <string.h>
3
4  void konversiKeRomawi(int nilai, char hasil[]) {
5      // Tabel angka dan angka romawi yang sesuai
6      int nilaiArab[] = {1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1};
7      char *angkaRomawi[] = {"M", "CM", "D", "CD", "C", "XC", "L", "XL", "X", "IX", "V", "IV", "I"};
8
9      hasil[0] = '\0'; // inisialisasi string hasil
10
11     for (int i = 0; i < 13; i++) {
12         while (nilai >= nilaiArab[i]) {
13             strcat(hasil, angkaRomawi[i]);
14             nilai -= nilaiArab[i];
15         }
16     }
17 }
18
19 int main() {
20     int angka;
21     char romawi[100]; // array untuk menampung hasil konversi
22
23     printf("Masukkan bilangan bulat positif: ");
24     scanf("%d", &angka);
25
26     if (angka <= 0) {
27         printf("Input harus bilangan positif!\n");
28     } else {
29         konversiKeRomawi(angka, romawi);
30         printf("Angka Romawi: %s\n", romawi);
31     }
32
33     return 0;
34 }
35
```

b. Pseudocode

**1. Algoritma :** Konversi Bilangan ke Romawi (1-10)

**Definisi Variabel:**

- integer angka

**Rincian Langkah:**

1. Cetak "Masukkan bilangan bulat (1 - 10): "
2. Baca angka
3. Cetak "Angka Romawi: "
4. **Switch** angka **do**
  - **Case 1:** Cetak "I" dan break
  - **Case 2:** Cetak "II" dan break
  - **Case 3:** Cetak "III" dan break
  - **Case 4:** Cetak "IV" dan break
  - **Case 5:** Cetak "V" dan break
  - **Case 6:** Cetak "VI" dan break
  - **Case 7:** Cetak "VII" dan break
  - **Case 8:** Cetak "VIII" dan break
  - **Case 9:** Cetak "IX" dan break
  - **Case 10:** Cetak "X" dan break
  - **Default:** Cetak "Input tidak valid!"
5. **EndSwitch**

**2. Algoritma :** Konversi Bilangan ke Romawi (Versi 2)

**Definisi Variabel:**

- integer angka
- string romawi

**Rincian Langkah:**

1. *Cetak "Masukkan bilangan bulat positif: "*
2. *Baca angka*
3. *If angka  $\leq 0$  then*
  - *Cetak "Input harus bilangan positif!"*
4. *Else*
  - *Panggil prosedur Konversi ke Romawi dengan parameter angka dan romawi*
  - *Cetak "Angka Romawi: ", romawi*
5. *EndIf*

8. Dalam bidang pengolahan citra (image processing), elemen gambar terkecil disebut pixel (picture element). Nilai pixel untuk gambar 256 warna adalah dari 0 sampai 255. Operasi-operasi terhadap pixel seringkali berada di luar rentang nilai ini. Jika ini kasusnya, maka nilai hasil operasi harus dipotong (clipping) sehingga tetap berada di dalam interval [0..255]. Jika nilai hasil operasi lebih besar dari 255, maka nilai tersebut dipotong menjadi 255, dan bila negatif maka dipotong menjadi 0. Dibaca sebuah nilai hasil operasi pengolahan citra, buatlah algoritma untuk melakukan clipping tersebut.

**Jawab:**

a. *Source Code*

```
1  #include <stdio.h>
2
3  int main() {
4      int nilaiPixel;
5
6      printf("Masukkan nilai hasil operasi pengolahan citra: ");
7      scanf("%d", &nilaiPixel);
8
9      // Clipping nilai ke rentang [0, 255]
10     if (nilaiPixel > 255) {
11         nilaiPixel = 255;
12     } else if (nilaiPixel < 0) {
13         nilaiPixel = 0;
14     }
15
16     printf("Nilai pixel setelah clipping: %d\n", nilaiPixel);
17
18     return 0;
19 }
20
```

*b. Pseudocode*

**Algoritma :** *Clipping Nilai Pixel*

**Definisi Variabel:**

- *integer nilaiPixel*

**Rincian Langkah:**

1. *Cetak "Masukkan nilai hasil operasi pengolahan citra: "*
2. *Baca nilaiPixel*
3. ***If nilaiPixel > 255 then***
  - *Set nilaiPixel = 255*
4. ***Else if nilaiPixel < 0 then***
  - *Set nilaiPixel = 0*
5. ***EndIf***
6. *Cetak "Nilai pixel setelah clipping: ", nilaiPixel*

- **Soal Pola Bintang**

Buatlah program yang menampilkan pola segitiga bintang sebagai berikut:

a. Pola 1

```
  *
 ***
*****
*****
*****
```

**Jawab:**

a. *Source Code*

```

1  #include <stdio.h>
2
3  int main() {
4      int i, j, tinggi;
5
6      printf("Masukkan tinggi segitiga: ");
7      scanf("%d", &tinggi);
8
9      for (i = tinggi; i >= 1; i--) {
10         for (j = tinggi; j > i; j--) {
11             printf(" ");
12         }
13         for (j = 1; j <= (2 * i - 1); j++) {
14             printf("*");
15         }
16         printf("\n");
17     }
18     return 0;
19 }

```

#### b. Flowchart

**Algoritma :** Pola Segitiga

**Definisi Variabel:**

- integer i, j, tinggi

**Rincian Langkah:**

1. Cetak "Masukkan tinggi segitiga: "
2. Baca tinggi
3. **For** i dari tinggi downto 1 **do**
  - **For** j dari tinggi downto (i + 1) **do**
    - Cetak " " (spasi tanpa pindah baris)
  - **EndFor**
  - **For** j dari 1 hingga (2 \* i - 1) **do**

- Cetak "\*" (tanpa pindah baris)

- EndFor
- Cetak baris baru

#### 4. EndFor

#### b. Pola 2

```
*****
*****
*****
***
*
```

Jawab:

##### a. Source Code

```
1  #include <stdio.h>
2
3  int main() {
4      int i, j, tinggi;
5
6      printf("Masukkan tinggi segitiga: ");
7      scanf("%d", &tinggi);
8
9      for (i = 1; i <= tinggi; i++) {
10         // for (j = tinggi; j > i; j--) {
11             //     printf(" ");
12             // }
13         for (j = tinggi; j >= (2 * i - 1); j--) {
14             printf("*");
15         }
16         printf("\n");
17     }
18
19     return 0;
20 }
```



*b. Flowchart*

**Algoritma :** Pola Segitiga Terbalik

**Definisi Variabel:**

- integer i, j, tinggi

**Rincian Langkah:**

1. Cetak "Masukkan tinggi segitiga: "
2. Baca tinggi
3. **For** i dari 1 hingga tinggi **do**
  - **For** j dari tinggi downto  $(2 * i - 1)$  **do**
    - Cetak "\*" (tanpa pindah baris)
  - **EndFor**
  - Cetak baris baru
4. **EndFor**

- **Soal Bilangan Fibonacci**

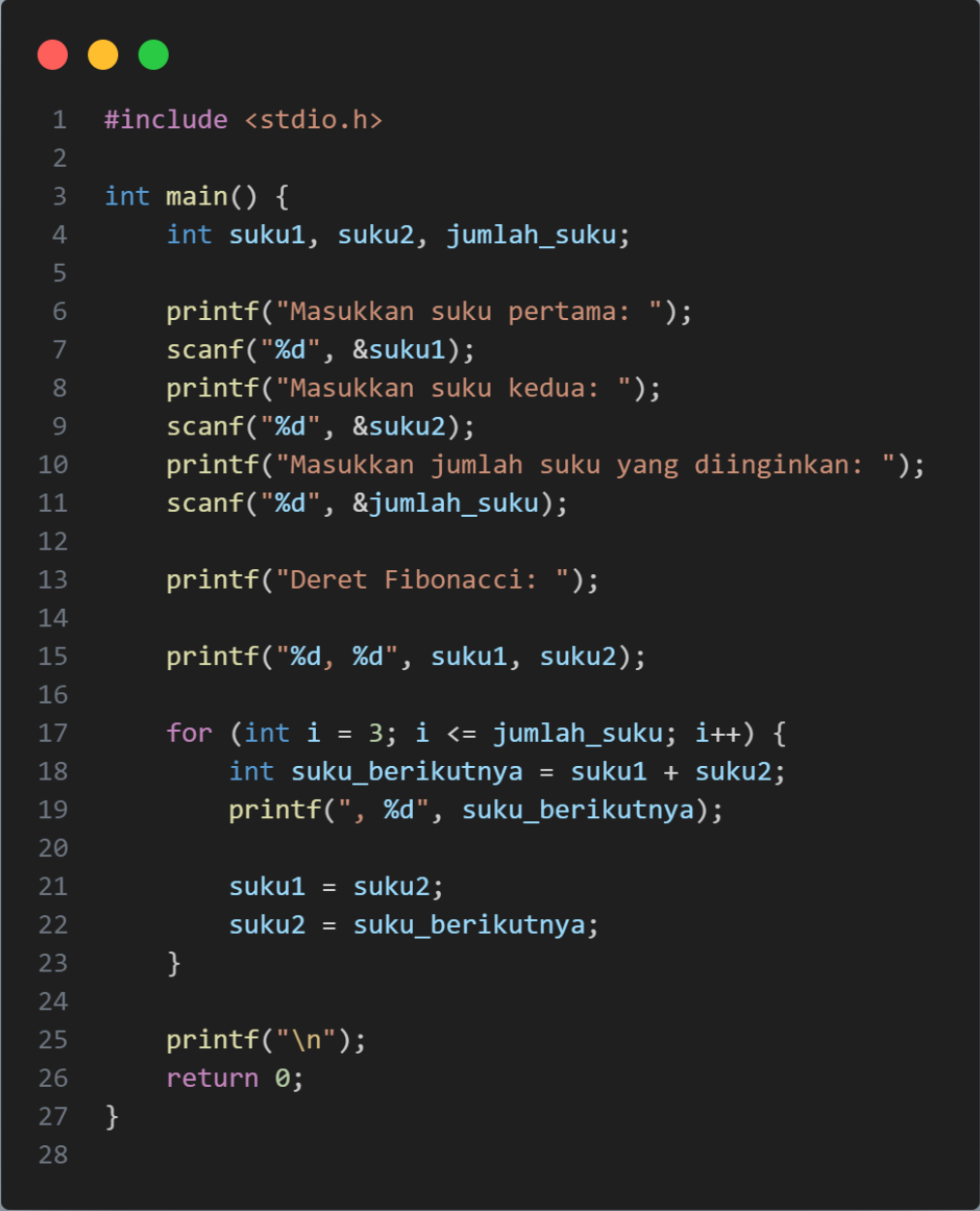
Buatlah program iteratif yang menampilkan dan menghitung deret Fibonacci yang suku pertama dan keduanya ditentukan oleh pengguna. Deret Fibonacci adalah deret angka yang dimana setiap angka berikutnya merupakan penjumlahan dari 2 angka sebelumnya. Contoh Fibonacci dimana suku pertama yaitu 2 dan suku kedua yaitu 3.

**2, 3, 5, 8, 13, ...**

Program harus meminta pengguna untuk memasukkan suku pertama, suku kedua, serta jumlah suku yang diinginkan dari deret Fibonacci. Program kemudian menampilkan deret Fibonacci hingga jumlah suku yang diminta.

**Jawab:**

a. Source Code



```
1  #include <stdio.h>
2
3  int main() {
4      int suku1, suku2, jumlah_suku;
5
6      printf("Masukkan suku pertama: ");
7      scanf("%d", &suku1);
8      printf("Masukkan suku kedua: ");
9      scanf("%d", &suku2);
10     printf("Masukkan jumlah suku yang diinginkan: ");
11     scanf("%d", &jumlah_suku);
12
13     printf("Deret Fibonacci: ");
14
15     printf("%d, %d", suku1, suku2);
16
17     for (int i = 3; i <= jumlah_suku; i++) {
18         int suku_berikutnya = suku1 + suku2;
19         printf(", %d", suku_berikutnya);
20
21         suku1 = suku2;
22         suku2 = suku_berikutnya;
23     }
24
25     printf("\n");
26     return 0;
27 }
28
```

b. Flowchart

**Algoritma : Deret Fibonacci**

**Definisi Variabel:**

- integer suku1, suku2, jumlah\_suku, suku\_berikutnya, i

**Rincian Langkah:**

1. Cetak "Masukkan suku pertama: "
2. Baca suku1
3. Cetak "Masukkan suku kedua: "
4. Baca suku2
5. Cetak "Masukkan jumlah suku yang diinginkan: "
6. Baca jumlah\_suku
7. Cetak "Deret Fibonacci: ", suku1, ", ", suku2
8. **For** i dari 3 hingga jumlah\_suku **do**
  - Set suku\_berikutnya = suku1 + suku2
  - Cetak ", ", suku\_berikutnya
  - Set suku1 = suku2
  - Set suku2 = suku\_berikutnya
9. **EndFor**
10. Cetak baris baru