Laboratorium Informatyki
Programowanie w języku C
Ćwiczenie 2

1 Wstęp

Praktycznie każdy program wykonuje określone fragmenty kodu w zależności od zaistniałej sytuacji. Podejmowanie decyzji możliwe jest dzięki instrukcji warunkowej lub wyboru. Inną ważną cechą współczesnych języków programowania jest możliwość wielokrotnego wykonania danego kodu. Zadanie to jest realizowane dzięki instrukcjom pętli.

2 Decyzja

Decyzje w kodzie pisanym w języku C można oprogramować za pomocą instrukcji warunkowej `if()` i wyboru `switch..case`. W języku C jest jeszcze dostępny operator wyboru `?` będący skrócona wersją instrukcji `if()`.

2.1 Instrukcja warunkowa

Instrukcja warunkowa `if()` wykonuje powiązany z nią kod wtedy gdy zdefiniowany w niej warunek zwróci wartość `true`. Instrukcja ta może reagować także gdy zwraca wartość `false`, wykonując kod zdefiniowany po słowie kluczowym `else`. Za pomocą `if()` można budować kaskadę warunkową umieszczając kolejny stopień `if()` po instrukcji `else`.

```
if(warunek_1){
   kod wykonywany gdy true dla warunek_1;
}else if(warunek_2){
   kod wykonywany gdy true dla warunek_2;
}else{
   kod wykonywany gdy false dla warunek_2;
}
```

Warunek jest funkcją logiczną budowaną zazwyczaj za pomocą operatorów relacji i logicznych: ==,!=, <, >, <=, >=, && (AND), || (OR) i ! (NOT).

Operator warunkowy `?` stosuje się w zastępstwie instrukcji warunkowej `if()` gdy w wyniku działania instrukcji warunkowej mają być wykonane proste wyrażenia podstawienia, obliczeniowe.

```
warunek ? wyrażenia_TRUE : wyrażenie_FALSE
```

Analogicznie jak w przypadku instrukcji warunkowej za pomocą operatora warunkowego także można budować układy kaskadowe.

```
#include <stdio.h>
int main()
{
    int x=76,y=120;
    printf("Liczba %i jest większa\n",(x>y) ? x: y);
}
```

2.2 Instrukcja wyboru

Instrukcja wyboru stosowana jest w zastępstwie instrukcji warunkowej gdy konieczne by było budowanie dla niej długiej kaskady. Instrukcja wyboru realizuje wskazany przez wartość parametru sterującego fragment kodu.

```
   switch(zmienna){
case wartość_1:
   kod;
   breal;
case wartość_2:
   kod;
   break;
default:
   kod;
}
```

Instrukcja `switch()` w zależności od wartości jaką przyjmie parametr `zmienna` realizuje przeskok w kodzie do miejsca gdzie po słowie `case` występuje żądana wartość. Jeżeli poszukiwana wartość nie jest powiązana z żadną z instrukcji `case` to dochodzi do przeskoku do linii `default`. Istotnym elementem każdego bloku `case` jest instrukcja `break` umieszczana na końcu bloku, a powodująca wyjście z bloku instrukcji `switch()`.

3 Powtarzanie kodu

Konstrukcje pętli pozwalają na wykonanie określonego fragmentu kodu zadaną liczbę razy. Kryterium decydującym o wyborze pętli do realizacji określonego zadania

jest sposób w jaki w kodzie będzie określana liczba powtórzeń. Jeżeli z góry wiadomo ile powtórzeń należy wykonać wybierana jest pętla iteracyjna `for()`, jeżeli kolejne wykonanie kodu pętli jest uzależnione od bieżącego stanu programu wybierane są pętle warunkowe `while` lub `do..while`.

3.1 Pętla iteracyjna

Pętla iteracyjna `for()` pozwala na precyzyjne określenie liczby wykonanych iteracji. Konstrukcja jest sterowana parametrem deklarowanym dla pętli. Pętla `for()` definiowana jest przez trzy parametry.

```
for(inicjacja;warunek powtórzenia;inkrementacja){
   kod pętli;
}
```

- **Inicjacja**: deklaracja zmiennej sterującej z określeniem jej wartości początkowej. Zmienna ta będzie dostępna tylko wewnątrz kodu powiązanego blokiem z pętlą `for()`. Zmienna sterująca może być zadeklarowana wcześniej przed pętlą, a w pętli `for()` określa się jej wartość początkową.
- **Warunek powtórzenia**: warunek logiczny sprawdzający określone kryterium dla zmiennej sterującej. Jeżeli warunek zwróci wartość `true` wykonany zostanie kod powiązany z pętlą.
- **Inkrementacja**: określa w jaki sposób po każdym wykonaniu kodu pętli zmieni się wartość parametru sterującego. Zazwyczaj wykorzystywana konstrukcja inkrementacji (`i++`) lub dekrementacji (`i--`).

3.2 Pętle warunkowe

W języku C dostępne są dwie konstrukcje pętli warunkowej, sprawdzająca warunek wykonania na początku (`while(){}`) i na końcu (`do{}while();`) pętli. Powtórzenie kodu pętli w obu przypadkach ma miejsce gdy warunek logiczny zdefiniowany w instrukcji `while()` zwróci wartość `true`.

```
while(i<10){
   printf("%i ",i);
   i++;
   }
```

```
do{
   printf("%i ",i);
   i++;
   }while(i<10);
```

Warunek powtórzenia pętli jest powiązany z realizowanym przez nią kodem. W przypadku pętli warunkowych istnieje ryzyko realizacji pętli nieskończonej. Istotne jest więc takie zapisanie kodu aby uniknąć tego ryzyka lub zabezpieczenie kodu instrukcją przerwania pętli.

3.3 Przerwanie pętli

Każdy blok instrukcji może zostać przerwany. W języku C dostępne są trzy instrukcje przerywające wykonywanie kodu. Poznana przy omawianiu instrukcji wyboru funkcja `break` przerywa działanie pętli. Wykonywanie kodu rozpocznie się od pierwszej linii po kodzie pętli. Instrukcja `continue` przerwie aktualną iterację pętli i rozpocznie kolejną. Pętlę można także przerwać instrukcją zwrotu funkcji `return`. Spowoduje ona jednak poza przerwaniem działania pętli także zakończenie aktualnej funkcji. Wszystkie instrukcje przerwania działają w powiązaniu z instrukcją warunkową `if()`.

3.4 Instrukcja skoku

Instrukcja skoku została odziedziczona przez C z wcześniejszych języków (BASIC). Działa ona podobnie jak instrukcja wyboru wykonując przeskok do wskazanej etykiety. Instrukcja skoku `goto nazwa;` działa w bezpośrednim powiązaniu z instrukcją warunkową `if()`. Miejsce skoku określa się za pomocą etykiety, dowolnej nazwy zakończonej znakiem dwukropka. Skok może być realizowany zarówno w górę jak i w dół kodu.

```
   kod;
nazwa1:
   kod;
   if(warunek) goto nazwa2;
   kod;
   if(warunek) goto nazwa1;
nazwa2:
   kod;
```

Zadania

Napisać programy realizujące podane poniżej zadania wykorzystując materiał omówiony w niniejszej instrukcji. W miarę potrzeb poszczególne zadania realizowane przez kod wydzielić do funkcji.

I Napisać program obliczający pierwiastki rzeczywiste równania kwadratowego.

II Napisać program wprowadzone przez użytkownika działanie matematyczne. Program umożliwia wykonanie działań (+,-,*,/,dowolnej potęgi, pierwiastka dowolnego stopnia). Program rozpoznaje czynniki i rodzaj wprowadzonego działania.

III Napisać program wyświetlający tabliczkę mnożenia dla zadanego przez użytkownika przedziału liczb. Zadanie wykonać wszystkimi rodzajami instrukcji pętli (trzy programy lub trzy funkcje).