Jazyk C: DU1

```
Jazyk C                   DU1                          22.2.2011
----------------------------------------------------------------


                     Domácí úkol č.1


Termín odevzdání:  22.3.2011
 čtěte pokyny na konci tohoto textu
 Hodnocení max. 15 bodů

Příklady: (budou opravovány v prostředí Linux/GCC,
           LC_ALL=cs_CZ.iso-8859-2
           parametry překladu -std=c99 -Wall -pedantic)

 Napište modul "error.c" s rozhraním v "error.h", který definuje
   funkci void Error(const char *fmt, ...). Tato funkce má
   stejné parametry jako printf(); tiskne text "CHYBA: " a potom
   chybové hlášení podle formátu fmt. Vše tiskne do stderr
   (funkcí vfprintf) a potom ukončí program voláním funkce exit(1).
   Použijte definice ze stdarg.h.

   Napište modul "enum.c" s rozhraním "enum.h",
   ve kterém definujete výčtový typ:
      enum months { ChybnyMesic=0, Leden=1, Unor, Brezen, Duben, Kveten,
           Cerven, Cervenec, Srpen, Zari, Rijen, Listopad, Prosinec };
   a funkce pro
     - tisk hodnoty výčtového typu do standardního výstupu:
          void PrintMonthShort(enum months d);
       tato funkce tiskne řetězec odpovídající zadané hodnotě
       výčtového typu česky ("Led", "Úno", "Bře", "Dub" ...).
          void PrintMonth(enum months d);
       tato funkce tiskne řetězec odpovídající zadané hodnotě
       výčtového typu ("Leden", "Únor", ...).
       V případě hodnoty parametru mimo  rozsah tiskne
       chybu  "PrintMonth*: Hodnota  %d je mimo rozsah\n"
       a to vámi definovanou funkcí Error.

     - čtení hodnoty výčtového typu ze standardního vstupu
          enum months ReadMonth(void);
       funkce přeskočí všechny bílé znaky (isspace),
       potom přečte slovo (posloupnost znaků z množiny definované makrem
       isalpha, zapněte si lokalizaci),
       potom provede převod ("Led", "led", "Leden", "LEden", ... na Leden)
       a vrátí výsledek. Pokud nejde o identifikaci měsíce, vrací nulu (ChybnyMesic).
       Funkce bude "case insensitive" a rozpozná všechny varianty vypisované
       funkcí strftime pro formát "%b" a "%B" (předpokládáme českou
```

```
        lokalizaci:  LC_TIME=cs_CZ date +"%b" -d now+1month ).
        Navíc rozpozná a převede varianty zapsané bez diakritiky - "cesky".

        Funkce musí používat statická pole řetězců definujících
        převod (v názvech měsíců uvažujte "cestinu" i diakritiku).

        Do poznámky na první řádek modulu napište
        // Encoding: ISO-8859-2
        nebo zkratku pro jiné vámi použité kódování podle IANA:
        http://www.iana.org/assignments/character-sets

    Napište testovací program "enumtest.c", kde ve funkci main
      - nejdříve načtěte ze std vstupu jednu hodnotu a vytiskněte
      - potom vypisujte měsíce do výstupního souboru
        dokud nedojde k chybovému ukončení programu:

        int main(void) { /* test - NEMĚNIT! */
          char *l = setlocale(LC_ALL,"cs_CZ.iso-8859-2");
          if(l==NULL)
              Error("setlocale: Nelze nastavit českou lokalizaci\n");
          enum months m;
          m = ReadMonth();     // čte měsíc
          PrintMonthShort(m);  // tiskne krátké jméno
          printf("\n");
          PrintMonth(m);       // tiskne dlouhé jméno
          printf("\n\n");
          for( m = Leden; m < 15; m++ ) { // úmyslná chyba
            PrintMonthShort(m);
            printf("\n");
          }
          return 0;
        }

     Použijte program "make" pro překlad/sestavení programu.
     Testovací příkaz:  echo "Led" | ./enumtest

(7b)
```

```
b) Definujte makra pro pole bitů:
     BitArray(jmeno_pole,velikost)
       definuje a nuluje pole (POZOR: MUSÍ _INICIALIZOVAT_ bez ohledu
       na to, zda je pole statické nebo automatické/lokální!
       Vyzkoušejte obě varianty.)
       Př: BitArray(p,1000L);   // p = pole 1000 bitů
           BitArray(q,10000L);  // q = pole 10000 bitů

     SetBit(jmeno_pole,index,výraz)
       nastaví zadaný bit v poli na hodnotu zadanou výrazem
       (nulový výraz --> bit 0, nenulový výraz --> bit 1)
       Př: SetBit(p,20,1);
```

```
      GetBit(jmeno_pole,index)
        získá hodnotu zadaného bitu, vrací hodnotu 0 nebo 1
        Př: if(GetBit(p,i)==1) printf("1");
            if(!GetBit(p,i))   printf("0");

   Kontrolujte meze polí. V případě chyby volejte funkci
     Error("Index %ld mimo rozsah 0..%ld", (long)index, (long)mez).
   [zájemci si zkusí překlad na 64bitové platformě, kde
     sizeof(int)!=sizeof(long)]
   Pro implementaci použijte pole typu: unsigned long [].
   Implementace musí efektivně využívat paměť (využít každý
   bit pole až na posledních max X-1, pokud má unsigned long X bitů).

   Podmíněným překladem zajistěte, aby se při definovaném symbolu
   USE_INLINE místo těchto maker definovaly inline funkce
   všude kde je to možné (bez změn v následujícím testovacím příkladu!).
   (Pozor na nestandardní sémantiku inline funkcí v GCC starším než verze 4.3
   viz "http://gcc.gnu.org/c99status.html")
   USE_INLINE nesmí být definováno ve zdrojovém textu --
   překládá se s argumentem -D (gcc -DUSE_INLINE ...).

   Jako testovací příklad implementujte Eratosthenovo síto na
   výpočet posledních 20 prvočísel ze všech prvočísel od 2 do
   N=90000000 (90 milionů).
   Budete pravděpodobně potřebovat zvětšit limit velikosti zásobníku.
   Na Unix-like systémech použijte příkaz ulimit -a pro zjištění velikosti
   limitu a potom ulimit -s. Doporučuji nejdříve odladit pro N=100.

   Každé prvočíslo tiskněte na zvláštní řádek v pořadí
   vzestupném.  Netiskněte  nic  jiného  než  prvočísla (bude se
   automaticky  kontrolovat!).  Pro kontrolu správnosti prvočísel
   můžete použít program  factor (./prvocisla|factor).

   Program se musí jmenovat "prvocisla.c" !
   Příkaz "make" musí vytvořit obě varianty: prvocisla a prvocisla-inline
   (Při nesplnění podmínek: až 0 bodů.)

   Zařiďte, aby příkaz "make" bez parametrů vytvořil všechny spustitelné
   soubory.  Při změně kteréhokoli souboru musí přeložit jen změněný soubor a
   závislosti.  Pokud bude Makefile vypadat jako skript odečtou se 4b.

(8b)

Poznámky:  Eratosthenovo síto (přibližná specifikace):
   1) bitové pole  p  o rozměru N, index  i  nastavit na 2
   2) vybereme nejmenší index i, takový, že p[i]==0. Potom je
      i prvočíslo
   3) pro všechny násobky i nastavíme bit p[n*i] na 1
      ('vyškrtneme' násobky - nejsou to prvočísla)
   4) i++; dokud nejsme za sqrt(N), opakujeme 2 až 4
      (POZOR: sestavit s matematickou knihovnou, případně zapnout optimalizace)
   5) Výsledek: v poli  p  jsou na prvočíselných indexech hodnoty 0

   Efektivita výpočtu: cca 1.5s na Core2duo/2.8GHz/Linux64 (gcc -O2)
   Porovnejte efektivitu obou variant. (Zvídaví studenti si mohou vyzkoušet
```

```
   program acovea: http://www.coyotegulch.com/products/acovea/)
```

```
Předmět: Jazyk C                                  rev 19.2.2011

Obecné pokyny pro vypracování domácích úkolů

*  Pro úkoly v jazyce C používejte ISO C99 (soubory *.c)
   Použití nepřenositelných konstrukcí není dovoleno.

*  úkoly zkontrolujte překladačem například takto:
      gcc -std=c99 -pedantic -Wall priklad1.c
   místo gcc můžete použít i jiný překladač
!  (nebude-li úkol podle normy ISO C99, bude za 0 bodů!)
   v  souvislosti s tím napište do poznámky na začátku
   souboru jméno překladače, kterým byl program přeložen
   (implicitní je verze GNU C instalovaná na serveru merlin).

*  Programy  pište, pokud je to možné, do jednoho zdrojového
   souboru. Dodržujte předepsaná jména souborů.

*  Na začátek každého souboru napište poznámku, která bude
   obsahovat jméno, fakultu, označení příkladu a datum.

Příklad:
  // enum.c
  // Řešení IJC-DU1, příklad a), 12.3.2100
  // Autor: Vogon Jeltz, FIT
  // Přeloženo: gcc 4.4
  // popis příkladu - poznámky, atd

* Úkoly je nutné zabalit programem zip takto:
       zip xnovak99.zip *.c *.h Makefile

  Jméno xnovak99 nahradíte vlastním. ZIP neobsahuje adresáře.
  Každý si zkontroluje obsah ZIP archivu jeho rozbalením v prázdném adresáři
  a napsáním "make".

* Řešení se odevzdává elektronicky v IS FIT

* Posílejte pouze nezbytně nutné soubory -- ne *.EXE !

* Úkoly neodevzdané v termínu budou za 0 bodů.

* Opsané úkoly budou hodnoceny 0 bodů pro všechny zůčastněné
  a to bez výjimky (+ bonus v podobě návštěvy u disciplinární komise).
```

Poslední modifikace: 22. února 2011
Pokud naleznete na této stránce chybu, oznamte to dopisem na adresu

peringer AT fit.vutbr.cz