2. test z PJC Jméno studenta:_____

Pocet bodu: ----

Varianta 25: příklad 1 (5 bodů)

Napište funkci, která korektně vymaže z paměti prvek jednosměrně zřetězeného seznamu. Prvek bude identifikován indexem (první prvek má index 0). Funkce musí také ošetřit nastavení globální promenné, která ukazuje na začátek seznamu. Správně ošetřete chybové stavy. Struktura prvku bude vypadat takto:

```
struct tprvek {
 void *data;
 struct tprvek *next;
};

/* globální proměnná: */
struct tprvek *first;

/* například, nebo si upravte */
int vymaz(int n);
```

Varianta 25: příklad 2 (5 bodů)

Napište funkci, která provede konverzi textového souboru ve formátu UNIX do formátu DOS (tj. prevede všechny znaky '\n' na sekvenci "\n\r"). Funkce má dva parametry, a to jméno vstupního a výstupního souboru. Funkce vrací pocet znaku vstupního souboru, pokud probehla v porádku, jinak vrací -1. Ošetrete chyby pri práci se soubory. V prípade chyby, funkce vypíše chybu na standardní chybový výstup.

Varianta 25: příklad 3 (5 bodů)

Standardní knihovní funkce

```
void qsort( void * base, int nelem, int elsize, int (*compare)(const void*, const void*))
```

umožnuje algoritmem QuickSort vzestupne seradit zadané pole base o nelem prvcích, kde každý prvek má velikost elsize. Uživatelem zadaná funkce compare je volána pro porovnání prvku pole, jejím výsledkem je císlo <0, ==0 nebo >0 pro signalizaci menší, rovno a vetší. Navrhnete podobnou funkci bsort (), která bude radit pole sestupne a bude používat algoritmus BubbleSort. Navrhnete a implementujte funkci IntCompare(), která muže být použita na míste uživatelské funkce compare pro pole celých císel. Implemetujte uvedené funkce jako modul, uvedte presný obsah .c a .h souboru.

Varianta 25: příklad 4 (5 bodů)

Uvažujte obousmerne zretezený seznam hodnot typu double, definovaný následovne:

```
typedef struct Tprvek {
  Tprvek *Poslední, *dalsi;
  double data;
} Tprvek;

typedef struct Tseznam {
  Tprvek *prvni, *Poslední;
  int pocet;
} Tseznam;
```

Pokud je prvek poslední, má vyplnenou položku Poslední na NULL, pokud je prvek první, má vyplnenou položku dalsi na NULL. V prípade prázdného seznamu, jsou položky prvni a Poslední v seznamu nastaveny na NULL.

Napište funkci, která pridá (zatrídí) prvek do seznamu tak, aby seznam byl setrídený (od nejmenšího po nejvetší prvek). Prvek se ve funkci musí alokovat. Funkce je deklarována následovne:

```
void zatrid(Tseznam *seznam, double hodnota);
```

Varianta 25: příklad 5 (5 bodů)

pole
Definujte binarni matici 15x50 bitu (50 radku, 15 sloupcu). Napište funkci, která spocita a vrátí pocet radku, které obsahuji pouze nuly. Pri implementaci matice vemte duraz vice na velikost pouzite pameti nez na rychost vypoctu.

Varianta 25: příklad 6 (5 bodů)

void Swap(int *x,int *y);
Toto je hlavicka procedury, která pomoci volani odkazem prohodi obsahy
dvou promennych typu int;Doplnte telo teto procedury a

uvedte kratky usek kodu,který demonstruje jeji pouziti.

Varianta 25: příklad 7 (5 bodů)

Napište program, který seradi vstupni soubor(polozky typu long). K
dispozici je procedura sort(long *a,long n), která seradi

"n" hodnot s zacatkem definovanym pointerem "a" ulozenych ve vnitrni
pameti.Odhadnete slozitost (sort ma slozitost n*log n).

Varianta 25: příklad 8 (5 bodů)

Navrhnete datovou strukturu, která umožní uchovávat retezce do délky MAX_LENGTH
v binárním stromu. Navrhnete rozhraní funkce TraverseTree(), která bude mít
vstupem ukazatel na koren stromu a ukazatel na uživatelskou funkci int f(char *).
Funkce TraverseTree() pracuje tak, že zavolá uživatelskou funkci f() na
každý z retezcu obsažených ve stromu. Pokud funkce f() vrací nenulu, má se v
pruchodu stromem pokracovat, pokud vrací nulu, má se procházení stromu okamžite
ukoncit. Implemetujte funkci TraverseTree() a funkci CountNodes(), která muže
být použita na míste uživatelské funkce pro získání poctu prvku v zadaném
stromu. Implemetujte uvedené funkce jako modul, uvedte presný obsah .c a .h
souboru.

Varianta 25: příklad 9 (5 bodů)

Necht je deklarovan soubor file takto:

FILE *file;

které z nasledujicich výrazu jsou spravne:

(a) file = fopen("muj.txt","R")
(b) file = fopen("muj.txt","a+")
(c) ftell(file)
(d) fclose(file)

Varianta 25: příklad 10 (5 bodů)

Necht je deklarovan soubor file takto:

FILE *file;

které z nasledujicich výrazu jsou spravne:

(a) freopen(file,"w+")
(b) freopen("tvuj.txt","w+",file)
(c) fclose(file)
(d) ftell(file)
---

2. test z PJC Jméno studenta:_____

Pocet bodu: ----

Varianta 26: příklad 1 (5 bodů)

Napište program,který nacte 2 čísla typu int, které byly programu predany
pri startu na prikazove radce. Vhodne osetrete

chybove stavy (neciselny vstup, spatny pocet parameru).

Varianta 26: příklad 2 (5 bodů)

Implementujte standardní funkci bsearch(), která algoritmem binárního pulení
zjištuje, zda zadané vzestupne serazené pole base o nelem prvcích, každý o
velikosti elsize bajtu obsahuje prvek odpovídající klíci key. Vlastní porovnání
je realizováno voláním funkce compare (), jejíž adresa (ukazatel) je pri volání
zadán. Návratovou hodnotou funkce je ukazatel na prvek pole, který odpovídá
klíci, prípadne NULL pokud pole takový prvek neobsahuje.

Varianta 26: příklad 3 (5 bodů)

Program mát na vstupu dve císla n a k. Necht množina M={ 1, 2, ... n }. Program
vypíše všechny variace s opakováním k-té trídy z prvku množiny M, tj. všechny
posloupnosti délky k, kde na každém míste je prvek z M. Zajistete, aby se vypsaly
opravdu všechny možnosti a žádná se neopakovala.

Varianta 26: příklad 4 (5 bodů)

Implementujte standardní knihovní funkci strcat, která pripojí zadaný
retezec src (ukoncený znakem '\0') na konec zadaného cílového retezce
dest (také ukonceného znakem '\0') a vrátí ukazatel na zacátek
cílového retezce dest. Neošetrujte prípadné pretecení cílového pole
(predpokládejte, že volající funkce ví, kolik místa pro cílový
retezec vyhradila, a jak dlouhé retezce tedy smí spojovat).

```
char *strcat(char *dest, const char *src);
```

Varianta 26: příklad 5 (5 bodů)

Uvazme nasledujici deklaraci:

```
struct tnode { int x:4; unsigned :6; int z:4; };
```

Tato deklarace popisuje:
(a) strukturu tnode s polozkami x a z s pocatecni hodnotami 4 a 6
(b) uzivatelsky typ tnode s polozkami x a z s pocatecni hodnotami 4 a 6
(c) strukturu tnode s polozkami x a z s delkou 4 a 6 a dirou 6 bitu
(d) chybne zapsana deklarace struktury

Varianta 26: příklad 6 (5 bodů)

Mějme strukturu pro spojový seznam:

```
struct node {
  int key;
  struct node *dalsi;
}
```

a makro:

```
#define abc(x, y) (x->dalsi = y, x)
```

a kod:

```
struct node *x, *y;
...
x = abc(y, x);
```

Poslední řádek kodu
(a) přidá y za začátek seznamu, začínajícího v x
(b) přidá x za začátek seznamu, začínajícího v y
(c) způsobí chybu - seznam, na který ukazovalo x je nenávratně ztracen
(d) způsobí chybu - seznam, na který ukazovalo y je nenávratně ztracen

---

Varianta 26: příklad 7 (5 bodů)

Napište program, který seradi vstupni soubor(polozky typu long). K
dispozici je procedura sort(long *a,long n), která seradi

"n" hodnot s zacatkem definovanym pointerem "a" ulozenych ve vnitrni
pameti.Odhadnete slozitost (sort ma slozitost n*log n).

Varianta 26: příklad 8 (5 bodů)

Uvažujte seznam grafických objektu: obdélník (levým horním rohem a pravým dolním rohem),
kruh (zadán stredem a polomerem) a trojúhelník (zadán tremi body). Definujte datový typ bod
a grafický objekt. Pro datový typ grafický objekt použijte union (uvedomte si, že samotný union nestací).

Uvažujte pole techto grafických objektu (velikost pole je dána hodnotou nejaké celocíselné promenné).
Napište funkci, která vytiskne na standardní výstup tyto grafické objekty (tj. jejich parametry).

Varianta 26: příklad 9 (5 bodů)

Necht je deklarovan soubor file takto:

FILE *file;

které z nasledujicich výrazu jsou spravne:

(a) file = ftell()
(b) freopen(file,"w+")
(c) fopen(file,"r+")
(d) freopen("tvuj.txt","w+",file)

Varianta 26: příklad 10 (5 bodů)

Necht je deklarovan soubor file takto:

FILE *file;

které z nasledujicich výrazu jsou spravne:

(a) ftell(file)
(b) file = fopen("muj.txt","a+")
(c) fopen(file)
(d) fopen(file,"r+")
---

2. test z PJC Jméno studenta:_____

Pocet bodu: ----

Varianta 27: příklad 1 (5 bodů)

Napište funkci, která korektně vymaže z paměti prvek jednosměrně zřetězeného seznamu. Prvek bude identifikován indexem (první prvek má index 0). Funkce musí také ošetřit nastavení globální promenné, která ukazuje na začátek seznamu. Správně ošetřete chybové stavy. Struktura prvku bude vypadat takto:

```
struct tprvek {
 void *data;
 struct tprvek *next;
};

/* globální proměnná: */
struct tprvek *first;

/* například, nebo si upravte */
int vymaz(int n);
```

Varianta 27: příklad 2 (5 bodů)

Navrhnete a implementujte funkci FileCopy(), která umožní zkopírovat zadaný soubor na nové místo. Parametrem funkce je dále ukazatel na uživatelskou funkci, int progress (double ratio), která má být periodicky volána jako signalizace postupu kopírování. Parametrem funkce je desetinné císlo ratio, které udává procentuální cást dosud zkopírovaného souboru, je-li její návratová hodnota je nenulová, má se pokracovat v kopírování, je-li nulová, má se kopírování okamžite ukoncit. Ošetrete a vhodne signalizujte chybové stavy "chyba cteni" a "chyba zápisu". Implementujte funkci tak, aby parametr progress mohl být i NULL (tj. informace o postupu kopírování volající nepožaduje). Implemetujte uvedenou funkci jako modul, uvedte presný obsah .c a .h souboru.

Varianta 27: příklad 3 (5 bodů)

Napište funkci, která zmení všechny tabulátory na rádku v souboru za mezery. Funkce má dva parametry: retezec (reprezentující rádek v souboru) a celé císlo (reprezentující pocet znaku na tabulátor). Funkce zmení znaky v zadaném retezci (tj. nealokuje pamet pro nový retezec), predpokládá se, že je alokováno dostatek místa pro vložení mezer v retezci.

Uvedomte si, že pocet vkládaných mezer je závislý na pozici tabulátoru na rádku, napr. tabulátor na 1. pozici v rádu se mení na 8 mezer (pri 8-mi mezerách na tabulátor) a na 2. pozici se mení za 7 mezer.

Varianta 27: příklad 4 (5 bodů)

Implementujte standardní knihovní funkci strcat, která pripojí zadaný retezec src (ukoncený znakem '\0') na konec zadaného cílového retezce dest (také ukonceného znakem '\0') a vrátí ukazatel na zacátek cílového retezce dest. Neošetrujte prípadné pretecení cílového pole (predpokládejte, že volající funkce ví, kolik místa pro cílový retezec vyhradila, a jak dlouhé retezce tedy smí spojovat).

```
char *strcat(char *dest, const char *src);
```

Varianta 27: příklad 5 (5 bodů)

Zapiste deklaraci struktury midd, která umozni vyuzit prostredni 4 bity spodni slabiky celeho čísla jako čísla bez znamenka.

Varianta 27: příklad 6 (5 bodů)

Mějme strukturu pro spojový seznam:

```
struct node {
  int key;
  struct node *dalsi;
}
```

a makro:

```
#define abc(x, y) (x->dalsi = y, x)
```

a kod:
struct node *x, *y;
...
x = abc(y, x);

Poslední řádek kodu
(a) přidá y za začátek seznamu, začínajícího v x
(b) přidá x za začátek seznamu, začínajícího v y
(c) způsobí chybu - seznam, na který ukazovalo x je nenávratně ztracen
(d) způsobí chybu - seznam, na který ukazovalo y je nenávratně ztracen
---
Varianta 27: příklad 7 (5 bodů)

Je dan vstupni soubor "obcane.txt" v nemz jednotlive radky maji strukturu
"jmeno prijmeni rodne_číslo" (rodne_číslo je ve

tvaru xxxxxx/xxxx). Vytvorte prg, který z techto obcanu vybere (do souboru
"vybrane.txt" ulozi) může narozene v roce 1970 a

jejichz krestni jmeno zacina na A, B, C nebo D.

Varianta 27: příklad 8 (5 bodů)

Definice

```
typedef struct node {
  int key;
  node *next;
} node;
```

je

a) zcela spravna
b) spravna, ale dvojznacna pri pouziti "node"
c) chybna, nazvy "node" a "struct node" se prekryvaji
d) chybna, neznamy typ pro clena "next" struktury

Varianta 27: příklad 9 (5 bodů)

Necht je deklarovan soubor file takto:

```
FILE *file;
```

které z nasledujicich výrazu jsou spravne:

(a) fopen(file)
(b) file = ftell()
(c) fopen(file,"r+")
(d) file = fopen("muj.txt","R")

Varianta 27: příklad 10 (5 bodů)

Necht je deklarovan soubor file takto:

```
FILE *file;
```

které z nasledujicich výrazu jsou spravne:

(a) freopen(file,"w+")
(b) freopen("tvuj.txt","w+",file)
(c) fclose(file)
(d) ftell(file)
---

2. test z PJC Jméno studenta:_____

Pocet bodu: ----

Varianta 28: příklad 1 (5 bodů)

Napište funkci, která korektně vymaže z paměti prvek obousměrně zřetězeného seznamu. Prvek bude identifikován indexem (první prvek má index 0). Funkce musí také ošetřit nastavení globálních proměnných, které ukazují na seznam (jedna na jeho první prvek a druhá na konec). Správně ošetřete chybové stavy. Struktura prvku bude vypadat takto:

```
struct tprvek {
 void *data;
 struct tprvek *next,*prev;
};

/* globální proměnné: */
struct tprvek *first,*last;

/* například, nebo si upravte */
int vymaz(int n);
```

Varianta 28: příklad 2 (5 bodů)

Napište fuknci, která bude pracovat jako printf, ale bude reagovat pouze na %d, %s, a %c ve formatovacim retezci. Funkce může mit deklaraci
int mujprintf(char *fmt, ...);

Varianta 28: příklad 3 (5 bodů)

Uvažujte binární strom definovaný následovne:

```
typedef struct Tuzel {
  Tuzel *levy, *pravy;
  double data;
} Tuzel;
```

Napište funkci, která uloží tento strom do binárního souboru. Navrhnete takovou strukturu souboru, aby bylo možné strom po prectení znovu (shodne) postavit. Funkce je deklarovaná takto:

```
int uloz_strom(char *name, Tuzel *strom);
```

kde name je jméno souboru a strom je ukazatel na koren stromu. Funkce vrací 1, pokud vše probehlo v porádku, jinak vrací 0 a vypisuje chybu na standardní chybový výstup.

Varianta 28: příklad 4 (5 bodů)

Napište funkci, která provede konverzi textového souboru ve formátu UNIX do formátu DOS (tj. prevede všechny znaky '\n' na sekvenci "\n\r"). Funkce má dva parametry, a to jméno vstupního a výstupního souboru. Funkce vrací pocet znaku vstupního souboru, pokud probehla v porádku, jinak vrací -1. Ošetrete chyby pri práci se soubory. V prípade chyby, funkce vypíše chybu na standardní chybový výstup.

Varianta 28: příklad 5 (5 bodů)

Uvazme nasledujici deklaraci:

```
struct tnode { int x:4; unsigned y:6; };
```

Tato deklarace popisuje:
(a) typ struktury tnode s polozkami x a y s pocatecni hodnotami 4 a 6
(b) typ struktury tnode s polozkami x a y s delkou 4 a 6
(c) uzivatelsky typ tnode s polozkami x a y s pocatecni hodnotami 4 a 6
(d) uzivatelsky typ tnode s polozkami x a y s delkou 4 a 6

Varianta 28: příklad 6 (5 bodů)

Predpokladejte nasledujici deklarace:
char lin[] = "beta", *ptr = lin;
a fragment programu:
(*ptr++)--; (*ptr++) -= 'e'-'l'; *ptr += 'f'-'t';
Jaka bude hodnota pole lin?
(a) "alfa"
(b) "beta"
(c) nedefinovana

(d) nebude obsahovat ukonceny retez

Varianta 28: příklad 7 (5 bodů)

Necht je dan textovy soubor s hexadecimalnimi čísly v rozsahu 0000 az FFFF. Kazde číslo zabira prave 4 znaky, tj. mala čísla maji na pocatku nulu, napr. 00F1. čísla jsou oddelena prave jednou mezerou. čísla jsou v souboru setridena podle velikosti. Implementujte funkci

```
int najdi(char *filename, unsigned int key);
```

která vrátí nulu prave tehdy, když soubor filename neobsahuje číslo key, jinak vrátí nenulovou hodnotu.

Varianta 28: příklad 8 (5 bodů)

Navrhnete datovou strukturu pro implementaci zásobníku hodnot typu integer. Implementace by mela být efektivní, ale nemenla by nijak dopredu omezovat velikost uložitelných dat, tj. velikost zásobníku by mela být pridelována podle skutecných požadavku a velikosti pameti. Dále by implementace mela být efektivní a neplýtvat zbytecne pametí napr. pro ukazatele spojového seznamu. Navrhnete a implementujte funkce StackInit(), StackPush() a StackPop(), které slouží pro inicializaci, zápis a ctení ze zásobníku. Ošetrete a vhodne signalizujte chybové stavy "ctení z prázdného zásobníku" a "nedostatek pameti".Implemetujte uvedené funkce jako modul, uvedte presný obsah .c a .h souboru.

Varianta 28: příklad 9 (5 bodů)

Necht je deklarovan soubor file takto:

```
FILE *file;
```

které z nasledujicich výrazu jsou spravne:

(a) fopen(file)
(b) file = ftell()
(c) fopen(file,"r+")
(d) file = fopen("muj.txt","R")

Varianta 28: příklad 10 (5 bodů)

```
char *str;
```

Jaký je rozdil mezi alokaci
```
str = malloc(10 * sizeof(char));
```
a
```
str = calloc(10, sizeof(char));
```

a) žádný
b) malloc vyhradi vždy souvisly kus pameti, zatimco calloc může vyhradit vice mensich casti (setri pamet)
c) v inicializaci prvku pole
d) calloc vyhradi sizeof(sizeof(char)) == sizeof(int), tedy nespravnou velikost
---

2. test z PJC Jméno studenta:_____

Pocet bodu: ----

Varianta 29: příklad 1 (5 bodů)

Napište funkci, která provede setřídění jednosměrně zřetězeného seznamu
podle položky data prvku seznamu. Použít můžete libovolný řadící
algoritmus. Ošetřete seřazování prázdného seznamu. Struktura prvku seznamu
bude vypadat takto:

```
struct tprvek {
 int data;
 struct tprvek *next;
};

/* například, nebo si upravte */
int serad(struct tprvek *seznam);
```

Varianta 29: příklad 2 (5 bodů)

Standardní knihovní funkce

```
void qsort( void * base, int nelem, int elsize, int (*compare)(const void*, const void*))
```

umožnuje algoritmem QuickSort vzestupne
seradit zadané pole base o nelem prvcích, kde každý prvek má velikost elsize.
Uživatelem zadaná funkce compare je volána pro porovnání prvku pole, jejím
výsledkem je císlo <0, ==0 nebo >0 pro signalizaci menší, rovno a vetší.
Navrhnete podobnou funkci bsort (), která bude radit pole sestupne a bude
používat algoritmus BubbleSort. Navrhnete a implementujte funkci IntCompare(),
která muže být použita na míste uživatelské funkce compare pro pole celých
císel. Implemetujte uvedené funkce jako modul, uvedte presný obsah .c a .h
souboru.

Varianta 29: příklad 3 (5 bodů)

Napište funkci, která setrídí pole komplexních císel podle velikosti jejich absolutní hodnoty.
Datový typ komplexní císlo je definováno následovne:

```
typedef struct Tcplx {
  double im, re
} Tcplx;
```

Funkce je deklarována takto:

```
void setrid(Tcplx *pole, int pocet);
```

kde parametr pocet je pocet prvku v poli. Pro trídení použijte (pokud možno) standardní funkci qsort().

Varianta 29: příklad 4 (5 bodů)

Implementujte funkci Day2YDay, která pro trojici den, mesic, rok vypocte, kolikaty den
v roce to byl. Predpokladame gregoriansky kalendar, tedy kalendar který bezne
pouzivame. Funkce vrátí poradi dne v roce, tedy napr: Day2YDay (1, 1, 2000) da 0,
Day2YDay (1, 3, 2000) da 60, Day2YDay (1, 3, 2001) da 59.

Varianta 29: příklad 5 (5 bodů)

Uvazme nasledujici deklaraci:

```
struct tnode { int x:4; unsigned :6; int z:4; };
```

Tato deklarace popisuje:
(a) typ struktury tnode s polozkami x a z s pocatecni hodnotami 4 a 6
(b) uzivatelsky typ tnode s polozkami x a z s pocatecni hodnotami 4 a 6
(c) typ struktury tnode s polozkami x a z s delkou 4 a 4 a dirou 6 bitu
(d) chybne zapsana deklarace struktury tnode

Varianta 29: příklad 6 (5 bodů)

Mějme strukturu pro spojový seznam:

```
struct node {
  int key;
  struct node *dalsi;
}
```

a makro:
#define abc(x, y) (x->dalsi = y, x)

a kod:
struct node *x, *y;
...
x = abc(y, x);

Poslední řádek kodu
(a) přidá y za začátek seznamu, začínajícího v x
(b) přidá x za začátek seznamu, začínajícího v y
(c) způsobí chybu - seznam, na který ukazovalo x je nenávratně ztracen
(d) způsobí chybu - seznam, na který ukazovalo y je nenávratně ztracen
---
Varianta 29: příklad 7 (5 bodů)

Funkce int open(const char *pathname, int flags); otevira soubor, který
ma buffer
a) nastavitelny funkci setbuf
b) nastavitelny funkci setvbuf
c) pevne dany, nezmenitelny, velikost zavisi na implementaci jazyka C
d) žádný, bufferovani je zalezitosti operacniho systemu

Varianta 29: příklad 8 (5 bodů)

Uvažujte seznam grafických objektu: obdélník (levým horním rohem a pravým dolním rohem),
kruh (zadán stredem a polomerem) a trojúhelník (zadán tremi body). Definujte datový typ bod
a grafický objekt. Pro datový typ grafický objekt použijte union (uvedomte si, že samotný union nestací).

Uvažujte pole techto grafických objektu (velikost pole je dána hodnotou nejaké celocíselné promenné).
Napište funkci, která vytiskne na standardní výstup tyto grafické objekty (tj. jejich parametry).

Varianta 29: příklad 9 (5 bodů)

Necht je deklarovan soubor file takto:

FILE *file;

které z nasledujicich výrazu jsou spravne:

(a) freopen("tvuj.txt","w+",file)
(b) file = ftell()
(c) freopen(file,"w+")
(d) file = fopen("muj.txt","R")

Varianta 29: příklad 10 (5 bodů)

Necht je deklarovan soubor file takto:

FILE *file;

které z nasledujicich výrazu jsou spravne:

(a) file = fopen("muj.txt","R")
(b) file = fopen("muj.txt","a+")
(c) freopen(file,"w+")
(d) file = ftell()
---

2. test z PJC Jméno studenta:_____

Pocet bodu: ----

Varianta 30: příklad 1 (5 bodů)

Proveďte správné přetypování tak, aby 4 byty v proměnné typu float se intepretovali jako 4 byty proměnné typu long.

```
float f;
long l;

l=/* přetypování */d;
```

Varianta 30: příklad 2 (5 bodů)

Implementujte funkci Day2YDay, která pro trojici den, mesic, rok vypocte, kolikaty den v roce to byl. Predpokladame gregoriansky kalendar, tedy kalendar který bezne pouzivame. Funkce vrátí poradi dne v roce, tedy napr: Day2YDay (1, 1, 2000) da 0, Day2YDay (1, 3, 2000) da 60, Day2YDay (1, 3, 2001) da 59.

Varianta 30: příklad 3 (5 bodů)

Napište funkci, která provede konverzi textového souboru ve formátu UNIX do formátu DOS (tj. prevede všechny znaky '\n' na sekvenci "\n\r"). Funkce má dva parametry, a to jméno vstupního a výstupního souboru. Funkce vrací pocet znaku vstupního souboru, pokud probehla v porádku, jinak vrací -1. Ošetrete chyby pri práci se soubory. V prípade chyby, funkce vypíše chybu na standardní chybový výstup.

Varianta 30: příklad 4 (5 bodů)

Vytvorte funkci pro vypocet faktorialu. Osetrete chybny vstup. Vyreste rekurzivnim zpusobem.

Varianta 30: příklad 5 (5 bodů)

Uvazme nasledujici deklaraci:

```
struct tnode { int x:4; unsigned y:6; };
```

Tato deklarace popisuje:
(a) strukturu tnode s polozkami x a y s pocatecni hodnotami 4 a 6
(b) strukturu tnode s polozkami x a y s delkou 4 a 6
(c) uzivatelsky typ tnode s polozkami x a y s pocatecni hodnotami 4 a 6
(d) uzivatelsky typ tnode s polozkami x a y s delkou 4 a 6

Varianta 30: příklad 6 (5 bodů)

Predpokladejte nasledujici deklarace:
char lin[] = "beta", *ptr = lin;
a fragment programu:
(*ptr++)--; (*ptr++) -= 'e'-'l'; *ptr += 'f'-'t';
Jaka bude hodnota pole lin?
(a) "alfa"
(b) "beta"
(c) nedefinovana
(d) nebude obsahovat ukonceny retez

Varianta 30: příklad 7 (5 bodů)

Napište program, který seradi vstupni soubor(polozky typu long). K dispozici je procedura sort(long *a,long n), která seradi

"n" hodnot s zacatkem definovanym pointerem "a" ulozenych ve vnitrni pameti.Odhadnete slozitost (sort ma slozitost n*log n).

Varianta 30: příklad 8 (5 bodů)

Navrhnete datovou strukturu pro implementaci zásobníku hodnot typu integer. Implementace by mela být efektivní, ale nemenla by nijak dopredu omezovat velikost uložitelných dat, tj. velikost zásobníku by mela být pridelována podle skutecných požadavku a velikosti pameti. Dále by implementace mela být efektivní a neplýtvat zbytecne pametí napr. pro ukazatele spojového seznamu. Navrhnete a implementujte funkce StackInit(), StackPush() a StackPop(), které slouží pro inicializaci, zápis a ctení ze zásobníku. Ošetrete a vhodne signalizujte chybové stavy "ctení z prázdného zásobníku" a "nedostatek pameti".Implemetujte uvedené

funkce jako modul, uvedte presný obsah .c a .h souboru.

Varianta 30: příklad 9 (5 bodů)

Necht je deklarovan soubor file takto:

FILE *file;

které z nasledujicich výrazu jsou spravne:

(a) freopen(file,"w+")
(b) freopen("tvuj.txt","w+",file)
(c) fopen(file,"r+")
(d) file = fopen("muj.dat","rb")

Varianta 30: příklad 10 (5 bodů)

Necht je deklarovan soubor file takto:

FILE *file;

které z nasledujicich výrazu jsou spravne:

(a) file = fopen("muj.txt","R")
(b) file = fopen("muj.dat","rb")
(c) fopen("muj.dat",file)
(d) file = ftell()

---

2. test z PJC Jméno studenta:_____

Pocet bodu: ----

Varianta 31: příklad 1 (5 bodů)

Napište funkci, která provede setřídění jednosměrně zřetězeného seznamu
podle položky data prvku seznamu. Použít můžete libovolný řadící
algoritmus. Ošetřete seřazování prázdného seznamu. Struktura prvku seznamu
bude vypadat takto:

```
struct tprvek {
 int data;
 struct tprvek *next;
};

/* například, nebo si upravte */
int serad(struct tprvek *seznam);
```

Varianta 31: příklad 2 (5 bodů)

Napište funkci, která orízne všechny bílé znaky na zacátku a na konci retezce. Pro zjištení
bílých znaku použijte makro isspace z hlavickového souboru ctype.h. Funkce bude mít jeden
parametr - predávaný retezec. Tento retezec bude také funkcí menen (tj. funkce nealokuje pamet).
Funkce vrací ukazatel na predávaný retezec.

Príklady:

```
" \n \r \t\t hello everybody \n\n" -> "hello everybody"
" \t        \n"                   -> ""
""                                -> ""
```

Varianta 31: příklad 3 (5 bodů)

Implementujte funkci Day2YDay, která pro trojici den, mesic, rok vypocte, kolikaty den
v roce to byl. Predpokladame gregoriansky kalendar, tedy kalendar který bezne
pouzivame. Funkce vrátí poradi dne v roce, tedy napr: Day2YDay (1, 1, 2000) da 0,
Day2YDay (1, 3, 2000) da 60, Day2YDay (1, 3, 2001) da 59.

Varianta 31: příklad 4 (5 bodů)

Vytvorte funkci pro vypocet faktorialu. Osetrete chybny vstup. Vyreste
rekurzivnim zpusobem.

Varianta 31: příklad 5 (5 bodů)

Uvazme nasledujici deklaraci:

```
struct tnode { int x:4; unsigned :6; int z:4; };
```

Tato deklarace popisuje:
(a) typ struktury tnode s polozkami x a z s pocatecni hodnotami 4 a 6
(b) uzivatelsky typ tnode s polozkami x a z s pocatecni hodnotami 4 a 6
(c) typ struktury tnode s polozkami x a z s delkou 4 a 4 a dirou 6 bitu
(d) chybne zapsana deklarace struktury tnode

Varianta 31: příklad 6 (5 bodů)

Mějme strukturu pro spojový seznam:

```
struct node {
  int key;
  struct node *dalsi;
}
```

a makro:

```
#define abc(x, y) (x->dalsi = y, x)
```

a kod:

```
struct node *x, *y;
...
x = abc(y, x);
```

Poslední řádek kodu
(a) přidá y za začátek seznamu, začínajícího v x
(b) přidá x za začátek seznamu, začínajícího v y
(c) způsobí chybu - seznam, na který ukazovalo x je nenávratně ztracen

(d) způsobí chybu - seznam, na který ukazovalo y je nenávratně ztracen
---
Varianta 31: příklad 7 (5 bodů)

Necht je dan textovy soubor s hexadecimalnimi čísly v rozsahu 0000 az FFFF. Kazde číslo zabira prave 4 znaky, tj. mala čísla maji na pocatku nulu, napr. 00F1. čísla jsou oddelena prave jednou mezerou. čísla jsou v souboru setridena podle velikosti. Implementujte funkci

int najdi(char *filename, unsigned int key);

která vrátí nulu prave tehdy, když soubor filename neobsahuje číslo key, jinak vrátí nenulovou hodnotu.

Varianta 31: příklad 8 (5 bodů)

Navrhnete vhodnou datovou strukturu, která umožní reprezentovat komplexní čísla bud v aritmetickém tvaru (rectangular, tj obsahuje složky real a imag), nebo v geometrickém tvaru (polar, tj. obsahuje složky abs a fi). Navržená struktura by nemela zbytecne plýtvat pametí. Navrhnete vhodné rozhraní funkcí Rec2Pol() a Pol2Rec(), které provádejí prevody mezi jednotlivými reprezentacemi. Implemetujte uvedené funkce jako modul, uvedte presný obsah .c a .h souboru.

Varianta 31: příklad 9 (5 bodů)

Necht je deklarovan soubor file takto:

FILE *file;

které z nasledujicich výrazu jsou spravne:

(a) file = ftell()
(b) freopen(file,"w+")
(c) fopen(file,"r+")
(d) freopen("tvuj.txt","w+",file)

Varianta 31: příklad 10 (5 bodů)

Necht je deklarovan soubor file takto:

FILE *file;

které z nasledujicich výrazu jsou spravne:

(a) fclose(file)
(b) file = fopen("muj.txt","R")
(c) fopen(file)
(d) fseek(file,0L,SEEK_END)
