**Structuri. Uniuni. Campuri de biti. Enumerari**

**I. Repere teoretice**

**Structuri**

* Definire – tip structurat care grupează date ce pot fi de tipuri diferite:

struct Nume

{ tip1 câmp1;

tip2 câmp2, câmp3;

...

} listă de variabile;

Observație: La declararea unei variabile de tip structura se aloca memorie pentru a putea memora toate campurile: sizeof(tip1)+sizeof(tip2)+....

Observație: Nume si lista de variabile pot lipsi, dar nu simultan.

Observație: dacă se precizează Nume, se pot face declarări ulterioare de variabile:

struct Nume listă de variabile;

typedef struct

{ tip1 câmp1;

tip2 câmp2, câmp3;

...

} NumeTip;

Observație: declarările ulterioare de variabile se pot face prin precizarea numelui tipului definit:

Nume listă de variabile;

* Operații specifice:
* Acces la câmpuri: variabilă.câmp

Observație: dacă p memorează adresa unei structuri, accesul la un câmp al structurii se face fie prin operatorul . ( (\*p).câmp ), fie prin operatorul -> (p->câmp )

* Atribuiri: premise la nivelul fiecărui camp și la nivelul întregii structuri

**Câmpuri de biți**

* Definire – tip structurat care permite accesul la un bit sau la un grup de biți:

struct Nume

{ tip1 câmp1:nrBiți1;

tip2 câmp2:nrBiți2, câmp3:nrBiți3;

...

} listă de variabile;

Observație: La declararea unei variabile de tip camp de biti se aloca memorie pentru a putea memora toate campurile: nrBiti1+nrBiti2+.... (rotunjit la un multiplu al unei unitati de memorie)

Observație: într-o structură pot coexista atât câmpuri de biți, cât și membri de tipuri clasice.

Observație: nu este necesar ca toți biții să fie atribuiți

Observație: nu este necesar ca toate câmpurile sa aiba nume.

Observație: Nume si lista de variabile pot lipsi, dar nu simultan.

Observație: dacă se precizează Nume, se pot face declarări ulterioare de variabile:

struct Nume listă de variabile;

typedef struct

{ tip1 câmp1:nrBiți1;

tip2 câmp2:nrBiți2, câmp3:nrBiți3;

...

} NumeTip;

Observație: declarările ulterioare de variabile se pot face prin precizarea numelui tipului definit:

Nume listă de variabile;

* Operații specifice:
* Acces la câmpuri: variabilă.câmp
* Atribuiri: premise la nivelul fiecărui camp și la nivelul întregii structuri

**Uniuni**

* Definire – tip structurat care memoreaza in aceeasi zona date cu semnificatii diferite:

union Nume

{ tip1 câmp1;

tip2 câmp2, câmp3;

...

} listă de variabile;

Observație: La declararea unei variabile de tip uniune se aloca memorie pentru a putea memora cel mai mare (din punct de vedere al spatiului ocupat) dintre campuri: maxim(sizeof(tip1), sizeof(tip2)....)

Observație: o variabila va memora fie campul1, fie campul2, fie campul3.... ; alegerea se face prin intermediul numelui campului.

Observație: Nume si lista de variabile pot lipsi, dar nu simultan.

Observație: dacă se precizează Nume, se pot face declarări ulterioare de variabile:

union Nume listă de variabile;

typedef union

{ tip1 câmp1;

tip2 câmp2, câmp3;

...

} NumeTip;

Observație: declarările ulterioare de variabile se pot face prin precizarea numelui tipului definit:

Nume listă de variabile;

* Operații specifice:
* Acces la câmpuri: variabilă.câmp
* Atribuiri: premise la nivelul fiecărui camp și la nivelul întregii structuri

**Enumerări**

* Definire – mulțime de constante de tip întreg care cuprinde toate valorile pe care le poate avea o variabilă de tipul respectiv

enum Nume{ constanta1, constanta2,...} listă de variabile;

Observație: prima constanta are valoarea 0 si fiecare constanta are valoarea constantei care o precede+1, sau o valoare impusa.

enum Nume{ constanta1, constanta2=val,...} listă de variabile;

Observație: La declararea unei variabile de tip enumerare se aloca memorie pentru a putea memora o valoare intreaga.

Observație: o variabila poate memora doar una dintre constantele enumerate la un moment dat

Observație: Nume si lista de variabile pot lipsi, dar nu simultan.

Observație: dacă se precizează Nume, se pot face declarări ulterioare de variabile:

enum Nume listă de variabile;

typedef enum { constanta1, constanta2,...} NumeTip;

Observație: declarările ulterioare de variabile se pot face prin precizarea numelui tipului definit:

Nume listă de variabile;

* Operații specifice:
* Atribuiri: premise la nivelul fiecărui camp și la nivelul întregii structuri

**II. Exerciții**

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | Fie declarările:  struct data  {  int zi, luna,an;  } d1,d2;  struct student  {  char nume[20];  struct data dataNasterii;  } e1,e2;  struct student grupa[16];  struct student \*p;  struct student \*gp;  Stabiliți efectul următoarelor expresii/instrucțiuni: | | |
|  | d1.zi | Acces la câmpul zi al variabilei d1  Variabila d1 este de tip structură.  Expresia d1.zi este de tip intreg | |
|  | e1.nume | Acces la câmpul **nume** al variabilei **e1**  Variabila e1 este de tip structură.  Expresia e1.nume este de tip sir de caractere | |
|  | e1.nume[0] | Acces la prima litera a numelui memorat in variabila e1  Variabila e1 este de tip structură.  Expresia e1.nume este de tip sir de caractere  Expresia e1.nume[0] este de tip char | |
|  | e1.dataNasterii | Acces la data nasterii memorata in variabila e1  Variabila e1 este de tip structură.  Expresia e1.dataNasterii este de tip structura | |
|  | e1.dataNasterii.zi | Acces la ziua nasterii memorata in variabila e1  Variabila e1 este de tip structură.  Expresia e1.dataNasterii este de tip structura  Expresia e1.dataNasterii.zi este de tip intreg | |
|  | grupa[3] | Acces la al 4-lea element al tabloului grupa  Variabila grupa este de tip tablou  Expresia grupa[3] este de tip structura | |
|  | grupa[3].nume | Acces la campul nume al celui de al 4-lea element al tabloului grupa  Variabila grupa este de tip tablou  Expresia grupa[3] este de tip structura  Expresia grupa[3].nume este de tip sir de caractere | |
|  | grupa[3].nume[0] | Acces la prima litera a numelui celui de al 4-lea element al tabloului grupa  Variabila grupa este de tip tablou  Expresia grupa[3] este de tip structura  Expresia grupa[3].nume este de tip sir de caractere  Expresia grupa[3].nume[0] este de tip char | |
|  | (\*p).nume | Acces la câmpul **nume** al structurii a carei adresa este memorata in variabila **p**  Expresia \*p este de tip structura  Expresia (\*p).nume este de tip sir de caractere | |
|  | p->nume | Acces la câmpul **nume** al structurii a carei adresa este memorata in variabila **p**  Expresia p->nume este de tip sir de caractere (echivalenta cu (\*p).nume) | |
|  | gp=(struct student\*)malloc(n\*sizeof(struct student)); | | |
|  |  | | Aloca memorie dinamic pentru un tablou de structuri |
| 2. | Fie declarările:  struct hotel  {  unsigned int camera:4;  unsigned int stare:1;  unsigned int achitat:1;  unsigned int nrLuni:2;  } h; | |  |
|  | Câți octeți ocupă variabila h? | | 1 octet |
|  | Care este valoarea maximă ce poate fi memorată în câmpul camera? | | 15 (0...24-1) |
|  | h.stare=1; | | Atribuie campului stare valoarea 1 (ocupat) |
| 3. | Fie declarările:  union studii  {  char \*NumeScoala;  struct  { char \*NumeLiceu;  double mediaBacalaureat;  } medii;  struct  {  char \*numeFacultate;  char \*profil;  char \*specializare;  int notaLicenta;  } superioare;  } st; | |  |
|  | Câți octeți ocupă variabila st? | | 16 octeti |
|  | st.medii.mediaBacalaureat=9.75; | | Atribuie campului mediaBacalaureat valoarea 9,75 |
| 4. | Fie declarările:  enum ziLucratoare  { luni, marti, joi=4, vineri  } zi; | |  |
|  | Câți octeți ocupă variabila zi? | | 4 octeti |
|  | Ce valoare are variabila zi după atribuirea de mai jos?  zi=luni; | | 0 |
|  | Ce valoare are variabila zi după atribuirea de mai jos?  zi=marti; | | 1 |
|  | Ce valoare are variabila zi după atribuirea de mai jos?  zi=vineri; | | 5 |

**III. Aplicații:**

1. Se memoreaza date despre studentii unei grupe:

nume, data nasterii, nota la cele 5 discipline pentru care sustin examen. Scrieti un program care, prin intermediul unui meniu

-citeste date pentru N studenti

-afiseaza datele tuturor studentilor

-afiseaza numele studentilor nascuti in anul x, citit de la tastatura;

-afiseaza numele studentului care are cea mai mare medie la disciplina x, citita de la tastatura.

Utilizati pentru implementare structuri imbricate (pentru data calendaristica si student) si tablouri de structuri.

2. Se citesc informatii despre mai multe figuri geometrice:

Tip figură (0-triunghi, 1-cerc, 2-dreptunghi)

Valori specifice: cele trei laturi ale unui triunghi sau raza cercului sau cele două laturi ale unui dreptunghi

Scrieti o aplicatie care determină aria și perimetrul fiecărei figuri.

Utilizați enumerări (pentru a defini cele trei tipuri de figuri geometrice), uniuni (pentru a memora cele trei tipuri de valori specifice ale figurilor), structuri (pentru a memora fiecare figura geometrica)

3. Se dau seturi de cate 3 numere, reprezentand cate o data calendaristica (zi, luna, an). Scrieti o aplicatie care, prin intermediul unui meniu, realizeaza operatii cu date calendaristice:

a) citeste informatii pentru n date calendaristice;

b) adauga o noua data calendaristica la multimea celor citite;

c) afiseaza toate datele citite;

d) sterge toate datele calendaristice din anul 2015;

e) ordoneaza datele cronologic (prin intermediul unui tablou).