## Algoritmos e Estruturas de Dados I (DCC/003)

### **Aula Prática 08**

- Instruções:
  - Os exercícios deverão ser feitos em aula de laboratório durante o tempo da aula;
  - o O professor/monitor irá esclarecer dúvidas em aula;
  - Crie uma pasta com seu nome e vá gravando seus programas implementados.
- **Para entrega**: submissão via moodle, conforme já explicado (veja instruções na própria página *Web* de submissão).

## Exercício 1 a 4

Estude os 4 programas, veja as suas diferenças, faça as alterações pedidas. Para cada alteração, gere um programa diferente, conforme pedido.

# Aprenda alterando um código já feito!

char descricao[40]; float valor; }; int main() {  struct produto prods[MAX]; /* definindo um vetor de 5 posicoes para o struc produto */ int i; for (i=0; i <max; (1)="" (2)="" (código="" (exemplo="" (i="0;" 0;="" 1)="" 120,00="" <max;="" [i="0;" acima="" altere="" ap08-ex01-1.c)="" buffer="" cep;");="" cidade="" com="" dados="" de="" digitados="" digitados.="" do="" em="" ender)="" ender.cidade_estado_cep);="" ender.cidade_estado_cep);<="" ender;="" endereco="" entre="" estado="" exibidos="" for="" foram="" getchar();="" gets(ender.rua);="" gets(prods[i].descricao);="" gets(prods[i].valor);="" gia="" i++)="" impresso="" imprime_endereco(struct="" inversa="" limpa="" na="" o="" obtem_endereco(void)="" ordem="" os="" para="" printf("\n\n");="" printf("\t="" produto?");="" produtos="" programa="" puts("nome="" que="" reais.="" return="" se\n",="" sejam="" somente="" struct="" teclado="" th="" valor="" void="" {="" }=""><th>Aprenda aiterando</th><th>um codigo ja feito!</th></max;>	Aprenda aiterando	um codigo ja feito!
struct produto {     char fescricao[40];     float valor;     };     struct produto prods[MAX]; /* definindo um vetor de 5     posicoes para o struc produto */     int main() {         struct produto prods[MAX]; /* definindo um vetor de 5     posicoes para o struc produto */     int int i;     for (i-0, i <max; (i-0,="" <max;="" do="" ender)="" ender.rua);="" ender.rua);<="" ender;="" endereco="" for="" gets(prods[i].descricao);="" i++)="" imprime_endereco(struct="" printf("\n\n");="" printf("\n\s\n",="" produto?");="" puts("nome="" puts("valor="" return="" th="" void="" {="" }=""><th>Exemplo 1</th><th>Exemplo 2</th></max;>	Exemplo 1	Exemplo 2
struct produto {     char descricao[40];     float valor;     };     int main() {     struct produto prods[MAX]; /* definindo um vetor de 5     posicoes para o struc produto */     int !;     for (=0; <max; (="0;" <max;="" buffer="" de="" descricao];="" do="" ender)="" ender.cidade_esta<="" ender.cidade_estado_cep);="" ender.rua);="" ender;="" endereco="" for="" getchar();="" gets(pender.ciadade_estado_cep);="" gets[cnds[],="" i++)="" imprime_endereco(struct="" limpa="" main(void)="" o="" printf("\k\s\n",="" printf("\n\n");="" printf("\n\s\n"),="" produto?");="" puts[nome="" puts[valor="" residencia;="" return="" struct="" td="" teclado="" void="" {="" }=""><td>#include <stdio.h></stdio.h></td><td>#define LEN 50</td></max;>	#include <stdio.h></stdio.h>	#define LEN 50
char rua[LEN]; char descricao[40]; float valor; }; int main() {  struct produto prods[MAX]; /* definindo um vetor de 5 posicoes para o struc produto */ int i; for (i=0; kMAX; i++) {  puts("Nome do Produto?");  gets[ender.rua];  printf("\t Entre cidade_estado_cep();  puts("Nalor do Produto?");  gets[ender.cidade_estado_cep);  printf("\t Entre cidade_estado_cep);  return ender; }  void imprime_endereco(struct endereco ender) {  printf("\t Nan, i++) {  printf("\t Sa\n", ender.rua);  printf("\t Sa\n", ender.cidade_estado_cep); }  void main(void) {  struct endereco ender) }  void imprime_endereco(struct endereco ender) {  printf("\t Sa\n", ender.cidade_estado_cep);  printf("\t Sa\n", en	#include <string.h></string.h>	
char cidade_estado_cep[LEN]; } struct produto f descricao[40]; float valor; };  struct produto prods[MAX]; /* definindo um vetor de 5 posicoes para o struc produto */ int i; puts["Nome do Produto?"); gets[prods[i].descricao:]; printf("\n\n"); for (i=0; kMAX; i++) {     printf("\n\n");     printf("\n\n");     printf("\n\n");     printf("\n\n");     printf("\n\n");     return 0; }  return 0; }  (1) Altere o programa (exemplo 1) para que os dados digitados sejam exibidos na ordem inversa em que foram digitados. (código ap08-ex01-1.c)  (2) Altere o programa (exemplo 1) para que seja impresso somente os produtos com valor acima de 120,00 reais.  char cidade_estado_cep[LEN]; } struct endereco obtem_endereco(void) {     struct endereco obtem_endereco ender;      printf("\t Entre rua: ");     gets[ender.cidade_estado_cep);     printf("\t %s\n", ender.rua);     printf("\t %s\n", ender.cidade_estado_cep);      void imprime_endereco(struct endereco ender) {     printf("\t %s\n", ender.rua);     printf("\t %s\n", ender.cidade_estado_cep); }  void main(void) {     struct endereco obtem_endereco ender;      printf("\t %s\n", ender.rua);     printf("\t %s\n", ender.cidade_estado_cep);      void main(void) {         struct endereco ender;          return ender; }  void imprime_endereco(struct endereco ender) {         printf("\t %s\n", ender.rua);         printf("\t %s\n", ender.cidade_estado_cep); }  void main(void) {         struct endereco ender;          return ender; }  void main(void) {         struct endereco ender;          printf("\t %s\n", ender.rua);         printf("\t %s\n", ender.cidade_estado_cep); }  void main(void) {         struct endereco ender;          return ender; }  void main(void) {         struct endereco ender;          printf("\t %s\n", ender.cidade_estado_cep);          printf("\t %s\n", ender.cidade_estado_cep); }  void main(void) {         struct endereco ender) {         printf("\t %s\n", ender.cidade_estado_cep); }          return ender; }  void main(void) {         struct endereco ender {	#define MAX 5	struct endereco {
char descricao[40]; float valor; }; int main() {  struct produto prods[MAX]; /* definindo um vetor de 5 posicoes para o struc produto */ int i; for (i=0; i <max; (1)="" (2)="" (código="" (exemplo="" (i="0;" 0;="" 1)="" 120,00="" <max;="" [i="0;" acima="" altere="" ap08-ex01-1.c)="" buffer="" cep;");="" cidade="" com="" dados="" de="" digitados="" digitados.="" do="" em="" ender)="" ender.cidade_estado_cep);="" ender.cidade_estado_cep);<="" ender;="" endereco="" entre="" estado="" exibidos="" for="" foram="" getchar();="" gets(ender.rua);="" gets(prods[i].descricao);="" gets(prods[i].valor);="" gia="" i++)="" impresso="" imprime_endereco(struct="" inversa="" limpa="" na="" o="" obtem_endereco(void)="" ordem="" os="" para="" printf("\n\n");="" printf("\t="" produto?");="" produtos="" programa="" puts("nome="" que="" reais.="" return="" se\n",="" sejam="" somente="" struct="" td="" teclado="" valor="" void="" {="" }=""><td></td><td>char rua[LEN];</td></max;>		char rua[LEN];
struct produto prods[MAX]; /* definindo um vetor de 5 posicoes para o struc produto */ init ; for (i=0; kMAX; i++) {     puts("Nome do Produto?");     scanf("%f", &prods[j].valor);     getchar(); // Limpa o buffer de teclado     }     printf("\n\n");     for (i=0; <max; (i="0;" <max;="" for="" i++)="" printf("\n\n");="" printf("\n\n");<="" td="" {=""><td>struct produto {</td><td>char cidade_estado_cep[LEN];</td></max;>	struct produto {	char cidade_estado_cep[LEN];
struct produto prods[MAX]; /* definindo um vetor de 5 posicoes para o struc produto */ int i; for (=0; <max; *="" *,="" *<="" do="" gets(prods[i].descricao);="" i++)="" printf("\t="" produto?");="" puts("nome="" td="" {=""><td></td><td>};</td></max;>		};
Struct produto prods[MAX]; /* definindo um vetor de 5   posicose para o struc produto */   init ; for [=0; k-MAX; i++) {	·	
struct produto prods[MAX]; /* definindo um vetor de 5 posicose para o struc produto */ int i; for ( =0;   <max; %s\n",="" &prods[i].valor);="" ( ="0;" buffer="" de="" do="" ender)="" ender.cidade_estado_cep);="" ender.rua);="" ender;="" endereco="" for="" getchar();="" gets(prods[i].descricao);="" imprime_endereco(struct="" limpa="" main(void)="" o="" printf("<="" printf("\n\n");="" printf("\t="" produto?");="" puts("nome="" puts("xelor="" scanf("%f",="" sets(ender.rua);="" struct="" td="" teclado="" void="" {=""  ++)=""  <max;="" }=""><td><b>}</b>;</td><td></td></max;>	<b>}</b> ;	
posicoes para o struc produto */ int i; for (i=0; kMAX; i++) {   puts("Nome do Produto?");   gets(prods[i].descricao);   puts("Valor do Produto?");   scanf("%f", &produto?");   scanf("%f", &produto?");   gets(ade_estado_cep);   printf("\n\n");   for (i=0; kMAX; i++) {     puts("Nome do Produto?");   scanf("%f", &produto?");   scanf("%f", &produto?");   scanf("%f", &produto?");   printf("\n\n");   for (i=0; kMAX; i++) {     puts("Valor do Produto?");   scanf("%f", &produto?");   scanf("%f", &produto?");   printf("\n\n");   for (i=0; kMAX; i++) {     puts("Nome do Produto?");   scanf("%f", &produto?");   printf("\n\n");   for (i=0; kMAX; i++) {     puts("\n")	int main() {	· ·
posicoes para o struc produto */ int i; for (i=0; kMAX; i++) {   puts("Nome do Produto?");   gets(prods[i].descricao);   puts("Valor do Produto?");   scanf("%f", &produto?");   scanf("%f", &produto?");   gets(ade_estado_cep);   printf("\n\n");   for (i=0; kMAX; i++) {     puts("Nome do Produto?");   scanf("%f", &produto?");   scanf("%f", &produto?");   scanf("%f", &produto?");   printf("\n\n");   for (i=0; kMAX; i++) {     puts("Valor do Produto?");   scanf("%f", &produto?");   scanf("%f", &produto?");   printf("\n\n");   for (i=0; kMAX; i++) {     puts("Nome do Produto?");   scanf("%f", &produto?");   printf("\n\n");   for (i=0; kMAX; i++) {     puts("\n")	struct produto prods[MAX]: /* definindo um vetor de 5	printf("\t Entre rua: "):
int i;     for (i=0; i <max; &prods[i].valor);="" (i="0;" <max;="" buffer="" de="" do="" for="" getchard);="" gets(prods[i].descricao);="" i++)="" i+-)="" limpa="" max;="" max;<="" o="" printf("\n\n");="" printf("\n\solor,="" produto?");="" puts("nome="" puts("valor="" scanf("%f",="" td="" teclado="" {="" }=""><td></td><td></td></max;>		
puts("Nome do Produto?"); gets(prods[i],descricao); puts("Valor do Produto?"); scanf("%f", &prods[i],valor); getchar(); // Limpa o buffer de teclado }  printf("\n\n"); for (i=0; <max; %.2f\n",="" %s="" %s\n",="" (exemplo="" (lembrar="" -="" 0;="" 1)="" :="" [10]="" [15]="" \0).<="" a="" altere="" caracter="" cep="" char="" cidade="" cidade,="" colocar="" como="" da="" dados="" de="" e="" ender)="" ender.cidade_estado_cep);="" ender.ini="" ender.rua);="" ender;="" endereco="" estado="" final="" i="" i++)="" imprime_endereco(struct="" main(void)="" mais="" o="" os="" para="" printf("\nseu="" printf("\t="" printf("bescricao;="" printf("entre="" prods[i].valor);="" programa="" que="" r\$="" recebidos="" residencia);="" residencia;="" return="" sejam="" seu="" string="" struct="" td="" ter="" um="" valor:="" void="" {="" }=""><td></td><td></td></max;>		
gets(prods[i].descricao); puts("Valor do Produto?"); scanf("%f", &prods[i].valor); getchar(); // Limpa o buffer de teclado } printf("\n\n"); for (i=0; <max; %.2f\n",="" %s="" %s\n",="" (1)="" (2)="" (código="" (exemplo="" (lembrar="" -="" 0;="" 1)="" 120,00="" 2)="" [10]="" [15]="" [3]="" \0).<="" a="" acima="" altere="" ap08-ex01-1.c)="" caracter="" cep="" char="" cidade="" cidade,="" colocar="" com="" como:="" da="" dados="" de="" digitados="" digitados.="" e="" em="" ender)="" ender.cidade_estado_cep);="" ender.rua);="" ender;="" endereco="" ensidencial:\n");="" estado="" exibidos="" final="" foram="" i++)="" impresso="" imprime_endereco(residencia);="" imprime_endereco(struct="" inversa="" main(void)="" mais="" na="" o="" ordem="" os="" para="" printf("\nseu="" printf("\t="" printf("descricao:="" prods[i].descricao,="" prods[i].valor);="" produtos="" programa="" que="" r\$="" reais.="" recebidos="" residencia;="" residincia="obtem_endereco();" return="" seja="" sejam="" somente="" string="" struct="" td="" ter="" um="" valor="" valor:="" void="" {="" }=""><td>for (i=0; i<max; i++)="" td="" {<=""><td>gets(ender.cidade_estado_cep);</td></max;></td></max;>	for (i=0; i <max; i++)="" td="" {<=""><td>gets(ender.cidade_estado_cep);</td></max;>	gets(ender.cidade_estado_cep);
puts("Valor do Produto?"); scanf("%f", &prods[i].valor); getchar(); // Limpa o buffer de teclado } printf("\n\n"); for (i=0; <max; %.2f\n",="" %s="" %s\n",="" (1)="" (2)="" (código="" (exemplo="" (lembrar="" -="" 0;="" 1)="" 120,00="" [10]="" [15]="" [3]="" \0).<="" a="" acima="" altere="" ap08-ex01-1.c)="" caracter="" cep="" char="" cidade="" cidade,="" colocar="" com="" como:="" da="" dados="" de="" digitados="" digitados.="" e="" eh:\n");="" em="" ender)="" ender.cidade_estado_cep);="" endereco="" estado="" exibidos="" final="" foram="" i++)="" impresso="" imprime_endereco(residencia);="" imprime_endereco(struct="" inversa="" main(void)="" mais="" na="" o="" ordem="" os="" para="" printf("\nseu="" printf("\t="" printf("descricao:="" prods[i].valor);="" produtos="" programa="" que="" r\$="" reais.="" recebidos="" residencia;="" return="" seja="" sejam="" somente="" string="" struct="" td="" ter="" um="" valor="" valor:="" void="" {="" }=""><td>puts("Nome do Produto?");</td><td></td></max;>	puts("Nome do Produto?");	
scanf("%f", &prods[i].valor); getchar(); // Limpa o buffer de teclado } printf("\n\n"); for (i=0; <max; %.2f\n",="" %s="" %s\n",="" (1)="" (2)="" (código="" (exemplo="" (lembrar="" -="" 0;="" 1)="" 120,00="" 2)="" [10]="" [15]="" [3]="" \0).<="" a="" acima="" altere="" ap08-ex01-1.c)="" caracter="" cep="" char="" cidade="" cidade,="" colocar="" com="" como:="" da="" dados="" de="" digitados="" digitados.="" e="" em="" en:\n");="" ender)="" ender.cidade_estado_cep);="" ender.rua);="" endereco="" estado="" exibidos="" final="" foram="" i++)="" impresso="" imprime_endereco(residencia);="" imprime_endereco(struct="" inversa="" main(void)="" mais="" na="" o="" ordem="" os="" para="" printf("\nseu="" printf("\t="" printf("descricao:="" prods[i].valor);="" produtos="" programa="" que="" r\$="" reais.="" recebidos="" residencia;="" return="" seja="" sejam="" somente="" string="" struct="" td="" ter="" um="" valor="" valor:="" void="" {="" }=""><td><pre>gets(prods[i].descricao);</pre></td><td>return ender;</td></max;>	<pre>gets(prods[i].descricao);</pre>	return ender;
yoid imprime_endereco(struct endereco ender)  {     printf("\n\n");     for (i=0; cMAX; i++) {         printf("Descricao: %s - Valor: R\$ %.2f\n",     prods[i].descricao, prods[i].valor);     }     return 0; }  return 0; }  (1) Altere o programa (exemplo 1) para que os dados digitados sejam exibidos na ordem inversa em que foram digitados. (código ap08-ex01-1.c)  digitados. (código ap08-ex01-1.c)  (2) Altere o programa (exemplo 1) para que seja impresso somente os produtos com valor acima de 120,00 reais.  void imprime_endereco(struct endereco ender)  {     printf("\t %s\n", ender.cidade_estado_cep);     printf("\t %s\n", ender.cidade_estado_cep);     printf("\t %s\n", ender.cidade_estado_cep);     printf("\t %s\n", ender.cidade_estado_cep); }  void main(void)  {     struct endereco ender)  {     printf("\t %s\n", ender.cidade_estado_cep); }  void main(void)  {     struct endereco(struct endereco ender)  {     printf("\t %s\n", ender.cidade_estado_cep); }  void main(void)  {     struct endereco ender)  {     printf("\t %s\n", ender.cidade_estado_cep); }  void main(void)  {     struct endereco ender)  {     printf("\t %s\n", ender.cidade_estado_cep); }  void main(void)  {     struct endereco ender)  {     printf("\t %s\n", ender.cidade_estado_cep); }  void main(void)  {     struct endereco ender)  {     printf("\t %s\n", ender.cidade_estado_cep); }  void main(void)  {     struct endereco ender.  {     printf("\t %s\n", ender.cidade_estado_cep); }  void main(void)  {     struct endereco en:\n");     imprime_endereco(residencia); }  printf("\nSeu endereco eh:\n");     imprime_endereco(residencia); }  Cl Altere o programa (exemplo 2) para que os dados cidade, estado e cep sejam recebidos como : Cidade - char [15]     Estado - char [10] (lembrar de colocar um caracter a mais para ter o final da string \( 0 \).	puts("Valor do Produto?");	}
Compared to the compared to		
printf("\n\n"); for (i=0; <max; %.2f\n",="" %s="" %s\n",="" (1)="" (2)="" (código="" (exemplo="" -="" 0;="" 1)="" 120,00="" acima="" altere="" ap08-ex01-1.c)="" com="" dados="" de="" digitados="" digitados.="" em="" ender.cidade_estado_cep);="" ender<="" exibidos="" foram="" i++)="" impresso="" inversa="" na="" o="" ordem="" os="" para="" printf("\k="" printf("descricao:="" prods[i].descricao,="" prods[i].valor);="" produtos="" programa="" que="" r\$="" reais.="" return="" seja="" sejam="" somente="" td="" valor="" valor:="" {="" }=""><td></td><td></td></max;>		
for (i=0; <max; %.2f\n",="" %s="" %s\n",="" (1)="" (2)="" (2p="" (código="" (exemplo="" (lembrar="" -="" 0;="" 1)="" 120,00="" 2)="" :="" [10]="" [15]="" [3]="" \0).<="" a="" acima="" altere="" ap08-ex01-1.c)="" caracter="" cep="" char="" cidade="" cidade,="" colocar="" com="" como="" da="" dados="" de="" digitados="" digitados.="" e="" eh:\n");="" em="" ender.cidade_estado_cep);="" endereco="" estado="" exibidos="" final="" foram="" i++)="" impresso="" imprime_endereco(residencia);="" inversa="" main(void)="" mais="" na="" o="" ordem="" os="" para="" printf("\t="" printf("descricao:="" prods[i].descricao,="" prods[i].valor);="" produtos="" programa="" que="" r\$="" reais.="" recebidos="" residencia;="" return="" seja="" sejam="" somente="" string="" struct="" td="" ter="" um="" valor="" valor:="" void="" {="" }=""><td>•</td><td>,</td></max;>	•	,
printf("Descricao: %s - Valor: R\$ %.2f\n", prods[i].descricao, prods[i].valor); } return 0; }  (1) Altere o programa (exemplo 1) para que os dados digitados sejam exibidos na ordem inversa em que foram digitados. (código ap08-ex01-1.c)  (2) Altere o programa (exemplo 1) para que seja impresso somente os produtos com valor acima de 120,00 reais.  }  void main(void) {     struct endereco residencia;     printf("Entre seu endereco eh:\n");     imprime_endereco(residencia); }  (1) Altere o programa (exemplo 2) para que os dados cidade, estado e cep sejam recebidos como:     Cidade - char [15]     Estado - char [3]     (Lembrar de colocar um caracter a mais para ter o final da string \0).		
prods[i].descricao, prods[i].valor); } return 0; }  struct endereco residencia;  printf("Entre seu endereco residencial:\n"); residencia = obtem_endereco();  printf("\nSeu endereco eh:\n"); imprime_endereco(residencia); }  (1) Altere o programa (exemplo 1) para que os dados digitados sejam exibidos na ordem inversa em que foram digitados. (código ap08-ex01-1.c)  (2) Altere o programa (exemplo 1) para que seja impresso somente os produtos com valor acima de 120,00 reais.  void main(void) {     struct endereco residencia;  printf("\nSeu endereco eh:\n"); imprime_endereco(residencia); }  (1) Altere o programa (exemplo 2) para que os dados cidade, estado e cep sejam recebidos como: Cidade - char [15] Estado - char [3] CEP - char [10] (lembrar de colocar um caracter a mais para ter o final da string \0).	* * * * * * * * * * * * * * * * * * * *	
void main(void) {     struct endereco residencia;      printf("Entre seu endereco residencial:\n");     residencia = obtem_endereco();      printf("\nSeu endereco eh:\n");     imprime_endereco(residencia); }  (1) Altere o programa (exemplo 1) para que os dados digitados sejam exibidos na ordem inversa em que foram digitados. (código ap08-ex01-1.c)  (2) Altere o programa (exemplo 1) para que seja impresso somente os produtos com valor acima de 120,00 reais.  void main(void) {     struct endereco residencia;  printf("\nSeu endereco eh:\n"); imprime_endereco(residencia); }  (1) Altere o programa (exemplo 2) para que os dados cidade, estado e cep sejam recebidos como: Cidade - char [15] Estado - char [3] CEP - char [10] (lembrar de colocar um caracter a mais para ter o final da string \0).		}
freturn 0; }  struct endereco residencia;  printf("Entre seu endereco residencial:\n"); residencia = obtem_endereco();  printf("\nSeu endereco eh:\n"); imprime_endereco(residencia); }  (1) Altere o programa (exemplo 1) para que os dados digitados sejam exibidos na ordem inversa em que foram digitados. (código ap08-ex01-1.c)  (2) Altere o programa (exemplo 1) para que seja impresso somente os produtos com valor acima de 120,00 reais.  {     struct endereco residencia;  printf("Entre seu endereco residencia:\n"); residencia = obtem_endereco();  printf("Entre seu endereco residencia:\n"); residencia = obtem_endereco();  printf("InSeu endereco eh:\n"); imprime_endereco(residencia); }  (1) Altere o programa (exemplo 2) para que os dados cidade, estado e cep sejam recebidos como: Cidade - char [15] Estado - char [3] CEP - char [10] (lembrar de colocar um caracter a mais para ter o final da string \0).		void main(void)
struct endereco residencia;  printf("Entre seu endereco residencial:\n"); residencia = obtem_endereco();  printf("\nSeu endereco eh:\n"); imprime_endereco(residencia); }  (1) Altere o programa (exemplo 1) para que os dados digitados sejam exibidos na ordem inversa em que foram digitados. (código ap08-ex01-1.c)  (2) Altere o programa (exemplo 1) para que seja impresso somente os produtos com valor acima de 120,00 reais.  struct endereco residencia;  printf("Entre seu endereco residencial:\n"); residencia = obtem_endereco();  printf("\nSeu endereco eh:\n"); imprime_endereco(residencial:\n"); residencia = obtem_endereco(n);  printf("\nSeu endereco eh:\n"); imprime_endereco(residencial:\n"); imprime_endereco(residencial:\n"); imprime_endereco eh:\n"); imprime_endereco(residencia);  (1) Altere o programa (exemplo 2) para que os dados cidade, estado e cep sejam recebidos como : Cidade - char [15] Estado - char [3] CEP - char [10] (lembrar de colocar um caracter a mais para ter o final da string \( 0 \)).	•	
residencia = obtem_endereco();  printf("\nSeu endereco eh:\n"); imprime_endereco(residencia); }  (1) Altere o programa (exemplo 1) para que os dados digitados sejam exibidos na ordem inversa em que foram digitados. (código ap08-ex01-1.c)  (2) Altere o programa (exemplo 1) para que seja impresso somente os produtos com valor acima de 120,00 reais.  (1) Altere o programa (exemplo2) para que os dados cidade, estado e cep sejam recebidos como: Cidade - char [15] Estado - char [3]  CEP - char [10] (lembrar de colocar um caracter a mais para ter o final da string \0).		,
printf("\nSeu endereco eh:\n"); imprime_endereco(residencia); }  (1) Altere o programa (exemplo 1) para que os dados digitados sejam exibidos na ordem inversa em que foram digitados. (código ap08-ex01-1.c)  (2) Altere o programa (exemplo 1) para que seja impresso somente os produtos com valor acima de 120,00 reais.  printf("\nSeu endereco eh:\n"); imprime_endereco(residencia);  (1) Altere o programa (exemplo 2) para que os dados cidade, estado e cep sejam recebidos como: Cidade - char [15] Estado - char [3]  CEP - char [10] (lembrar de colocar um caracter a mais para ter o final da string \0).		printf("Entre seu endereco residencial:\n");
(1) Altere o programa (exemplo 1) para que os dados digitados sejam exibidos na ordem inversa em que foram digitados. (código ap08-ex01-1.c)  (2) Altere o programa (exemplo 1) para que seja impresso somente os produtos com valor acima de 120,00 reais.  (1) Altere o programa (exemplo 2) para que os dados cidade, estado e cep sejam recebidos como:  Cidade - char [15]  Estado - char [3]  CEP - char [10]  (lembrar de colocar um caracter a mais para ter o final da string \0).		residencia = obtem_endereco();
(1) Altere o programa (exemplo 1) para que os dados digitados sejam exibidos na ordem inversa em que foram digitados. (código ap08-ex01-1.c)  (2) Altere o programa (exemplo 1) para que seja impresso somente os produtos com valor acima de 120,00 reais.  (1) Altere o programa (exemplo 2) para que os dados cidade, estado e cep sejam recebidos como:  Cidade - char [15]  Estado - char [3]  CEP - char [10]  (lembrar de colocar um caracter a mais para ter o final da string \0).		
que os dados digitados sejam exibidos na ordem inversa em que foram digitados. (código ap08-ex01-1.c)  (2) Altere o programa (exemplo 1) para que seja impresso somente os produtos com valor acima de 120,00 reais.  que os dados cidade, estado e cep sejam recebidos como:  Cidade - char [15]  Estado - char [3]  (Lembrar de colocar um caracter a mais para ter o final da string \0).		imprime_endereco(residencia);
que os dados digitados sejam exibidos na ordem inversa em que foram digitados. (código ap08-ex01-1.c)  (2) Altere o programa (exemplo 1) para que seja impresso somente os produtos com valor acima de 120,00 reais.  que os dados cidade, estado e cep sejam recebidos como:  Cidade - char [15]  Estado - char [3]  (Lembrar de colocar um caracter a mais para ter o final da string \0).		}
que os dados digitados sejam exibidos na ordem inversa em que foram digitados. (código ap08-ex01-1.c)  (2) Altere o programa (exemplo 1) para que seja impresso somente os produtos com valor acima de 120,00 reais.  que os dados cidade, estado e cep sejam recebidos como:  Cidade - char [15]  Estado - char [3]  (Lembrar de colocar um caracter a mais para ter o final da string \0).	(1) Altere o programa (exemplo 1) para	(1) Altere o programa (exemplo2) para
na ordem inversa em que foram digitados. (código ap08-ex01-1.c)  (2) Altere o programa (exemplo 1) para que seja impresso somente os produtos com valor acima de 120,00 reais.  recebidos como : Cidade - char [15] Estado - char [3] (Lembrar de colocar um caracter a mais para ter o final da string \0).		que os dados cidade, estado e cep seiam
digitados. (código ap08-ex01-1.c)  (2) Altere o programa (exemplo 1) para que seja impresso somente os produtos com valor acima de 120,00 reais.  Cidade - char [15] Estado - char [3]  (EP - char [10] (lembrar de colocar um caracter a mais para ter o final da string \0).	na ordem inversa em que foram	
Estado - char [3]  (2) Altere o programa (exemplo 1) para que seja impresso somente os produtos com valor acima de 120,00 reais.  Estado - char [3]  (lembrar de colocar um caracter a mais para ter o final da string \0).	<u>-</u>	Cidade - char [15]
que seja impresso somente os produtos com valor acima de 120,00 reais. (lembrar de colocar um caracter a mais para ter o final da string $\setminus 0$ ).	5 ( 8 F	2 2
com valor acima de 120,00 reais. para ter o final da string \0).	(2) Altere o programa (exemplo 1) para	CEP - char [10]
com valor acima de 120,00 reais. para ter o final da string \0).	que seja impresso somente os produtos	(lembrar de colocar um caracter a mais
	com valor acima de 120,00 reais.	para ter o final da string \0).
	(código ap08-ex01-2.c)	(código ap08-ex02.c)

Exemplo 3	Exemplo 4
#include <stdio.h></stdio.h>	#include <stdio.h></stdio.h>
#include <string.h></string.h>	#include <stdlib.h></stdlib.h>
struct tipo_endereco	#define TAM_VET 4
{	
char rua [50];	// declaracao do modelo data
int numero;	struct data{
char bairro [20];	int dia, mes, ano;
char cidade [30];	};
char sigla_estado [3];	// declaracao do modelo funcionario
long int CEP;	struct aluno{ char nome[50];
}; struct ficha_pessoal	double mediaFinal;
struct richa_pessoan	struct data nascimento;
char nome [50];	};
long int telefone;	
struct tipo_endereco endereco;	int main(){
};	int i;
main (void)	// declaracao e inicialização do vetor de estruturas vet
{	struct aluno faculdade[TAM_VET]={{"Adriana Martins", 7.5,
struct ficha_pessoal ficha;	{10,5,1982}},
strcpy (ficha.nome,"Luiz Osvaldo Silva");	{"Luciano Araujo de Lima", 8, {23,8,1983} },
ficha.telefone=4921234;	{"Marcos Paulo da Silva", 9.2, {14,1,1981}
strcpy (ficha.endereco.rua,"Rua das Flores");	},
ficha.endereco.numero=10;	{"Jonathas Martins dos Santos", 10,
strcpy (ficha.endereco.bairro, "Cidade Velha");	{2,10,1983}}
strcpy (ficha.endereco.cidade,"Belo Horizonte");	};
strcpy (ficha.endereco.sigla_estado,"MG");	puts("Imprimindo os dados.\n");
ficha.endereco.CEP=31340230;	for(i=0; i <tam_vet; i++){="" puts(faculdade[i].nome);<="" td=""></tam_vet;>
/* INICIDA AOLII AC LINILIAC DADA INADDINAID O	printf("Media Final: %.2f", faculdade[i].mediaFinal);
/* INSIRA AQUI AS LINHAS PARA IMPRIMIR O	printf("\n%d/%d\n\n", faculdade[i].media mai), printf("\n%d/%d\%d\n\n", faculdade[i].nascimento.dia,
QUE FOI COPIADO PARA A STRUCT FICHA	faculdade[i].nascimento.mes,
PESSOAL */	faculdade[i].nascimento.ano );
	}
	return (0);
	}
/* FIM DA AREA DE IMPRESSAO */	
/ THE DA ANEA DE HEFTELSSAO /	
return 0;	
}	
(1) Altere o programa exemplo 3 para	(1) Altere o programa exemplo 4 para
que se imprima o que foi copiado para	que a entrada de dados seja feita por
dentro da struct ficha pessoal.	meio de digitação e não como feito
	dentro do programa.
(código ap08-ex03.c)	
· · · ·	(código ap08-ex04.c)
	(

## Observações importantes:

- gets já é comando ultrapassado, assim recomendo utilizar fgets, que funciona assim:
  - o fgets (campo string, tamanho a ser lido, stdin);
  - Antes do comando fgets colocar um comando para limpar o buffer de entrada de dados:

```
    fflush (stdin); // no Windows
    fpurge (stdin); // no Mac
    __fpurge (stdin); // no Linux
```

#### Exercício 5

Vamos elaborar um programa em C para praticar os seguintes conceitos:

- Estrutura ou registro - struct

O que deverá ser feito é descrito a seguir:

Um hospital da rede pública de saúde deseja fazer o cadastro dos seguintes dados sobres os pacientes que atende: **nome**, **idade**, **peso** e **altura**. Defina uma estrutura de dados conveniente para armazenar estes dados (*struct*).

```
typedef struct pessoa {
    ...
    ...
} individuo;
```

Considere que o cadastro deverá ser armazenado em um **vetor** (denominado **vetorCadastro**), sendo que o vetor deverá ter um tamanho máximo permitido (#define MAX 10).

No início do código principal, pergunte ao usuário quantas pessoas serão cadastradas (n). Para cada pessoa a ser cadastrada, você receberá os dados e colocará no vetor do tipo da estrutura que você criou (individuo).

Em seu código main você deverá ter:

```
int main (int argc, char *argv[])
{
    individuo vetorCadastro[MAX];
    faça um estrutura de repetição para receber os cadastros das pessoas.

    faça um FOR (para cada pessoa), imprimindo seus campos:
        - vetorCadastro[i].nome;
        - vetorCadastro[i].altura;
        - etc.
}
```

Salve o código como ap08-ex05.c.