

```

1
2 char v[16][20]={ // declaracao de array global
3     "",//0
4     "Aracaju",//1
5     "Belém",//2
6     "Belo Horizonte",//3
7     "Brasília",//4
8     "Florianópolis",//5
9     "Fortaleza",//6
10    "Goiânia",//7
11    "João Pessoa",//8
12    "Natal",//9
13    "Porto Alegre",//10
14    "Recife",//11
15    "Rio de Janeiro",//12
16    "Salvador",//13
17    "São Luiz",//14
18    "São Paulo",//15
19 };
20
21 typedef struct Tdados{ //registro
22     int idia;//dia inicial do sistema
23     int imes;//mes inicial do sistema
24     int iano;//ano inicial do sistema
25     int dia;
26     int mes;
27     int ano;
28     char nome[50];
29     char data[11];
30     int origem;
31     int destino;
32     int contvoo;//contador de voo
33     int contespera;//contador de fila de espera
34     char lista[16][50]; //lista de passageiros para o voo e fila de espera
35 }Tdados;// registro de dados
36
37 Tdados IniciaTdados(){//metodo de inicializador
38     int i;
39     Tdados aux;
40     aux.idia=0;
41     aux.imes=0;
42     aux.iano=0;
43     aux.dia=0;
44     aux.mes=0;
45     aux.ano=0;
46     aux.contvoo=0;
47     aux.contespera=0;
48     strcpy(aux.nome,"");//atribuindo vazio a string
49     strcpy(aux.data,"");//atribuindo vazio a string
50     aux.origem=0;
51     aux.destino=0;
52
53     for(i=0;i<16;i++){//limpa de 0 a 15
54         strcpy(aux.lista[i],"");//atribuindo vazio a string de cada posicao do vetor
55     }
56     return aux;
57 }
58
59 void espacos(){ // pular linhas
60     int i;
61     for(i=0;i<5;i++){
62         printf("\n");
63     }
64 }
65
66 void listaaerportos(int exclui){//imprime a lista de aeroportos
67     int i;//variavel de contro do for
68     printf("-----\n");
69     for(i=1;i<=15;i++){//exclui recebe o valor de 1 a 15 de origem ou 0 se for usado para origem
70         if(i!=exclui){//imprime o conteudo do vetor de 1 a 15 que for diferente do valor de exclui
71             printf("%d %s \n",i,v[i]);
72         }
73     } //lista de aeroportos
74     printf("-----\n");
75 }
76
77 char *valornome(){//captura o valor de voo
78     espacos();// inserir alguns enter na tela
79     char nome[50];
80     printf("-----\n");
81     printf(" Nome do passageiro .: \n");
82     printf("-----\n");
83     fflush(stdin);//limpar buffer de entrada antes da leitura
84     gets(nome);//capturar variavel
85     strupr(nome);//deixar string maiuscula
86     fflush(stdin);//limpar buffer depois da leitura
87     espacos();//inserir alguns enter na tela
88
89     return nome;//retorna valor da funcao
90 }
91
92 char *validanome(){//validacao de nome
93     char nome[50];
94     int flag;
95     flag=0;
96     do{//repete enquanto o nome estiver VAZIO
97         system("cls");
98         if(flag==1){//se tiver vazio aparece essa mensagem
99             printf("\n\n\n-----\n");
100             printf(" !!! O CAMPO NOME NÃO PODE SER VAZIO !!!\n");
101             printf("-----\n");
102         }
103         strcpy(nome,valornome());//atribui o valor retornado da funcao para nome
104
105         if(strcmp(nome,"")==0){//se tiver vazio flag recebe 1
106             flag=1;
107         }else{//se nao tiver vazio flag recebe zero
108             flag=0;
109         }
110     }while(flag==1);//compara se o nome esta vazio enquanto for 1 repete
111
112     return nome;//retorno da funcao

```

```

113 }
114
115 int valororigem()//captura valor de origem
116 int origem;
117 listaaerportos(0);//exibe lista de aeroportos SEM exclusao de nomes
118 printf("-----\n");
119 printf(" Informe o numero correspondente a origem .: \n");
120 printf("-----\n");
121 scanf("%d",&origem);//captura o valor para origem
122 fflush(stdin);//limpar buffer depois da leitura
123 espacos();//inserir enter na tela
124
125 return origem;//retorno da funcao
126 }
127
128 int validaorigem()//validacao de origem
129 int valor,flag;
130 flag=0;
131 do{//repita ate que o valor seja entre 1 e 15
132
133     if(flag==1){//se repitiu aparece a mensagem ...
134         system("cls");
135         printf("\n\n-----\n");
136         printf(" !!! O CAMPO ORIGEM DEVE TER APENAS UM DOS NUMEROS LISTADOS !!!\n");
137         printf("-----\n");
138     }
139     valor=valororigem();//atribui valor retornado da funcao para a variavel valor
140
141     if(valor<1 || valor>15){//se o valor eh menor que 1 ou maior que 15 repete
142         flag=1;
143     }else{//senao sai do loop
144         flag=0;
145     }
146 }while(flag==1);
147 return valor;//retorno da funcao
148 }
149
150 int valordestino(int origem){//captura valor de destino
151 int destino,flag;
152 flag=0;
153 do{//REPITA enquanto DESTINO for igual a ORIGEM
154
155     if(flag==1){//se repitiu aparece a mensagem ...
156         system("cls");//limpa tela
157         printf("\n\n-----\n");
158         printf(" !!! O CAMPO DESTINO NÃO PODE SER IGUAL A ORIGEM !!!\n");
159         printf("-----\n");
160     }
161     listaaerportos(origem);//mostra lista de aeroportos excluindo a origem
162     printf("-----\n");
163     printf(" Informe o numero correspondente ao destino desejado .: \n");
164     printf("-----\n");
165     scanf("%d",&destino);//captura valor de destino
166     fflush(stdin);//limpar buffer depois da leitura
167     espacos();//inserir enter na tela
168
169     if(origem==destino){//se o valor de origem é igual a destino flag recebe
170         flag=1;
171     }else{//senao sai do loop
172         flag=0;
173     }
174
175 }while(flag==1);//repete enquanto origem for igual a destino
176 return destino;//retorno da funcao
177 }
178
179 int validadestino(int dest){//validacao de destino
180 int valor,flag;
181 flag=0;
182 do{//REPITA ate que destino seja entre 1 e 15
183     if(flag==1){//se repitiu aparece a mensagem ...
184         printf("\n\n-----\n");
185         printf(" !!! O CAMPO DESTINO DEVE TER APENAS UM DOS NUMEROS LISTADOS !!!\n");
186         printf("-----\n");
187     }
188
189     valor=valordestino(dest);//recebe retorno da funcao de destino
190     if(valor<1 || valor>15){//se o valor eh menor que 1 ou maior que 15 repete
191         flag=1;
192     }else{//senao sai do loop
193         flag=0;
194     }
195 }while(flag==1);//sai do loop somente quando for DIFERENTE de 1
196 return valor;//retorno da funcao
197 }
198
199 char *valordata()//captura data em formato string
200 char data[11];
201 printf("\nInforme a data desejada: ");
202 scanf("%s",data);
203 fflush(stdin); //limpar buffer de entrada
204 espacos(); //insere enter na tela
205 return data;
206 }
207
208 void stringtodata(Tdados *aux){//data em string
209 int i;
210 char data[11],dia[3],mes[3],ano[5];
211 printf("Enter a number: ");
212 fgets ( data, 11, stdin );//captura string (variavel, tamanho da string, comando )
213 dia[0]=data[0];//copia o caracter da posicao especifica de uma string para outra
214 dia[1]=data[1];//copia o caracter da posicao especifica de uma string para outra
215 dia[2]='\0';//adiciona nulo para a posicao da string
216 mes[0]=data[2];//copia o caracter da posicao especifica de uma string para outra
217 mes[1]=data[3];//copia o caracter da posicao especifica de uma string para outra
218 mes[2]='\0';//adiciona nulo para a posicao da string
219 ano[0]=data[4];//copia o caracter da posicao especifica de uma string para outra
220 ano[1]=data[5];//copia o caracter da posicao especifica de uma string para outra
221 ano[2]=data[6];//copia o caracter da posicao especifica de uma string para outra
222 ano[3]=data[7];//copia o caracter da posicao especifica de uma string para outra
223 ano[4]='\0';//adiciona nulo para a posicao da string
224 strcpy(data,dia);//copia dia para data

```

```

225     strcat(data, "/"); //concatena (adiciona, soma) a string o segundo valor imposto dentro dos parenteses
226     strcat(data, mes); //concatena (adiciona, soma) a string o segundo valor imposto dentro dos parenteses
227     strcat(data, "/"); //concatena (adiciona, soma) a string o segundo valor imposto dentro dos parenteses
228     strcat(data, ano); //concatena (adiciona, soma) a string o segundo valor imposto dentro dos parenteses
229     printf("%s", dia); //concatena (adiciona, soma) a string o segundo valor imposto dentro dos parenteses
230     aux->dia = atoi (dia); //converte inteiro para string
231     aux->mes = atoi (mes); //converte inteiro para string
232     aux->ano = atoi (ano); //converte inteiro para string
233
234     printf (" %d %d %d %s\n", aux->dia, aux->mes, aux->ano, data);
235 }
236
237 void mostradigitadodata(Tdados *aux, int flag){ //mostra data digitada até o momento
238     char v1[2], v2[2], v3[4]; //variaveis de auxilio
239     system("cls"); //limpa tela
240     espacos(); //coloca enter na tela
241
242     if(aux->dia<10){ strcpy(v1, "0"); } // se o dia é menor que 10 variavel de auxilio v1 recebe "0"
243     else{ strcpy(v1, ""); } // senao a variavel de auxilio v1 fica vazia
244
245     if(aux->mes<10){ strcpy(v2, "0"); } // se o mes é menor que 10 variavel de auxilio v2 recebe "0"
246     else{ strcpy(v2, ""); } // senao a variavel de auxilio v2 fica vazia
247
248     if(aux->ano<10){ strcpy(v3, "000"); } // se o ano é menor que 10 variavel de auxilio v3 recebe "000"
249     else{
250         if(aux->ano<100){ strcpy(v3, "00"); } // se o ano é menor que 10 variavel de auxilio v3 recebe "00"
251         else{
252             if(aux->ano<1000){ strcpy(v3, "0"); } // se o ano é menor que 10 variavel de auxilio v3 recebe "0"
253             else{ strcpy(v3, ""); } // senao a variavel de auxilio v3 fica vazia
254         }
255     }
256
257     printf(" DD/MM/AAAA \n");
258     if(flag==1){ //se a flag recebida por parametro no chamamento da funcao for 1 imprimi ...
259         printf(" / /%s%d \n", v3, aux->ano);
260     }
261     else {
262         if(flag==2){ //se a flag recebida por parametro no chamamento da funcao for 2 imprimi ...
263             printf(" /%s%d/%s%d \n", v2, aux->mes, v3, aux->ano);
264         }
265         else {
266             if(flag==3){ //se a flag recebida por parametro no chamamento da funcao for 3 imprimi ...
267                 printf(" %s%d/%s%d/%s%d \n", v1, aux->dia, v2, aux->mes, v3, aux->ano);
268             }
269         }
270     }
271
272 void vdata(Tdados voo[TAM], Tdados *aux){ //captura a data validada em inteiro
273
274 int flagano, flagmes, flagdia, igualmes, flag; //flags de validacao
275 int limitedia;
276 int r4, r100, r400;
277 char resp; //reposta que chama usada para a pergunta se digitou corretamente a hora
278 flagano=flagmes=flagdia=limitedia=igualmes=0; //inicializando variaveis
279
280 flag=0;
281 do{//REPITA enquanto ATE que ano seja MAIOR OU IGUAL que o ano ATUAL
282     if(flag==2){ //exibe a mensagem se nao passar na condicao
283         printf("\n\n\n-----\n");
284         printf(" ANO DIGITADO NÃO PODE SER MENOR QUE ANO ATUAL !!!\n");
285         printf("-----\n");
286         system("pause"); //pausa o programa para permitir que o usuario leia o aviso
287     }
288     printf("Informe o ANO .: ");
289     scanf("%d", &aux->ano); //leitura de dados
290     fflush(stdin); //limpar buffer apos a leitura
291     if(aux->ano>aux->iano){ //se ano fornecido for maior ou igual que o ano inicial
292         if(aux->ano>9999){
293             printf("-----\n");
294             printf("!!! O ANO DIGITADO NÃO ESTÁ NO FORMATO AAAA !!!\n");
295             printf("!!! O ANO DEVE TER APENAS 4 DIGITOS !!!\n");
296             printf("-----\n");
297             flag=3;
298         }
299         else{
300             flag=1; //.. entao flag recebe 1
301         }
302     }
303     }else{// senao se o ano digitado por menor que o atual
304         flag=2; //.. entao flag recebe 0
305     }
306 }while(flag==2 || flag==3); //repete o laço enquanto flag for 2
307
308 if(aux->ano==aux->iano){ //se ano digitado for IGUAL ao ano ATUAL
309     flagano=1;
310 }
311 flag=0;
312 do{//REPITA enquanto o Ano e mes for MENOR que o atual
313     mostradigitadodata(aux, 1); //mostra a data com apenas ano
314     printf("Informe o MES .: ");
315     scanf("%d", &aux->mes); //captura valor da variavel
316     if(flagano==1){ //se o ano digitado é igual ao ano atual
317         if(aux->mes>aux->imes){ //se o mes digitado for maior ou igual ao mes atual
318             if(aux->mes>0 && aux->mes<=12){ //se o mes digitado estiver no intervalo de 1 a 12
319                 flagmes=1;
320             }
321             else{//senao exibe uma mensagem ao usuario
322                 flagmes=0;
323                 flag=1;
324                 printf("\n\n\n-----\n");
325                 printf(" O MES DIGITADO NÃO ESTÁ NO INTERVALO DE 1 A 12\n");
326                 printf("-----\n");
327                 system("pause"); //pausa o programa para permitir que o usuario leia a mensagem
328             }
329         }
330         else{//se o ANO ATUAL for igual ao ano digitado e o MES ATUAL eh menor que o mes digitado
331             flag=1;
332             printf("\n\n\n-----\n");
333             printf(" O MES DIGITADO É MENOR QUE O MES ATUAL\n");
334             printf("-----\n");
335             if(!(aux->mes>0 && aux->mes<=12)){ //se alem disso estiver fora do intervalo .. exibir a mensagem
336                 printf("\n\n\n\t\t E \n");
337                 printf("\n\n\n-----\n");

```

```

337         printf("\ O MES DIGITADO NÃO ESTÁ NO INTERVALO DE 1 A 12\n");
338         printf("-----\n\n");
339     }
340
341     flagmes=0;
342     system("pause");//pausa o programa para permitir que o usuario leia a mensagem
343 }
344 }else{//se o ano digitado nao for igual ao atual ...
345     if ((flagano==0)
346         &&(aux->mes>0 && aux->mes<=12)){
347         flagmes=2;//se ano for maior verifica se o mes digitado esta entre 1 e 12
348     }else{//se nao estiver no intervalo exibe uma mensagem
349
350         printf("\n\n\n-----\n");
351         printf("\ O MES DIGITADO NÃO ESTÁ NO INTERVALO DE 1 A 12\n");
352         printf("-----\n");
353         system("pause");//pausa o programa para permitir que o usuario leia a mensagem
354     }
355 }
356
357 }while(!((flagmes==1) || (flagmes==2)));//repete ateh que flagmes seja igual a 1 ou 2
358
359 do{//REPITA ate que o dia digitado esteja entre o limite de dia que o mes digitado permite
360     mostradigitadodata(aux,2);//mostra data com mes e ano
361     printf("Informe o DIA .: ");
362     scanf("%d",&aux->dia);//captura o valor da variavel
363     switch(aux->mes)
364     {
365         case 1 : case 3 : case 5 : case 7 : case 8 : case 10 : case 12 :
366             limitedia=31;//caso o mes seja 1 3 5 7 8 10 12 o valor da variavel vira 31
367             if(aux->dia>0 && aux->dia<=limitedia){//se o dia digitado estiver no intervalo
368                 flagdia=1;
369             }else{//se nao estiver no intervalo possivel do mes entao exibe a mensagem
370                 flagdia=0;
371                 printf("\n\n\n-----\n");
372                 printf("\ O DIA DIGITADO NÃO ESTÁ NO INTERVALO DE 1 A %d\n",limitedia);
373                 printf("-----\n");
374                 system("pause");//pausa o programa para permitir que o usuario leia a mensagem
375             }
376             break;
377
378         case 4: case 6: case 9: case 11:
379             limitedia=30;//caso o mes seja 4 6 9 11 o valor da variavel vira 30
380             if(aux->dia>0 && aux->dia<=limitedia){
381                 flagdia=1;
382             }else{//se nao estiver no intervalo possivel do mes entao exibe a mensagem
383                 flagdia=0;
384                 printf("\n\n\n-----\n");
385                 printf("\ O DIA DIGITADO NÃO ESTÁ NO INTERVALO DE 1 A %d\n",limitedia);
386                 printf("-----\n");
387                 system("pause");//pausa o programa para permitir que o usuario leia a mensagem
388             }
389             break;
390
391         case 2:
392             r4=aux->ano % 4;
393             r100=aux->ano % 100;
394             r400=aux->ano % 400;
395             if((r4==0)&&(r100!=0)||(r400==0)){
396                 limitedia=29;
397                 if(aux->dia>0 && aux->dia<=limitedia){//se o dia digitado estiver no intervalo
398                     flagdia=1;
399                 }else{//se nao estiver no intervalo possivel do mes entao exibe a mensagem
400                     printf("\n\n\n-----\n");
401                     printf("\ O DIA DIGITADO NÃO ESTÁ NO INTERVALO DE 1 A %d\n",limitedia);
402                     printf("-----\n");
403                     system("pause");//pausa o programa para permitir que o usuario leia a mensagem
404                     flagdia=0;
405                 }
406             }
407             }else {
408                 limitedia=28;
409                 if(aux->dia>0 && aux->dia<=limitedia){//se o dia digitado estiver no intervalo
410                     flagdia=1;
411                 }else{//se nao estiver no intervalo possivel do mes entao exibe a mensagem
412                     printf("\n\n\n-----\n");
413                     printf("\ O DIA DIGITADO NÃO ESTÁ NO INTERVALO DE 1 A %d\n",limitedia);
414                     printf("-----\n");
415                     system("pause");//pausa o programa para permitir que o usuario leia a mensagem
416                     flagdia=0;
417                 }
418             }
419             break;
420     }
421     }//FIM DO SWITCH CASE
422     if(aux->mes==aux->imes){//se mes digitado for IGUAL o mes ATUAL
423         igualmes=1;
424     }else{ igualmes=0; }
425     //se o mes e ano for igual e o dia digitado eh menor que o dia atual exibe a mensagem
426     if(aux->dia< aux->idia && flagano==1 && igualmes==1){
427         printf("\n\n\n-----\n");
428         printf("\ O DIA DIGITADO É MENOR QUE O DIA ATUAL \n");
429         printf("-----\n");
430         system("pause");//pausa o programa para permitir que o usuario leia a mensagem
431         flagdia=0;
432     }
433 }while(!((flagdia==1)));//se tiver entre o limite nao repete
434 flag=0;
435 do{
436     //MOSTRA DATA DIGITADA ATEH AGORA
437     mostradigitadodata(aux,3);//mostra a data digitada
438
439     if(flag==1){// mostra a mensagem se o valor for diferente de S e N
440         printf("\n\n\n-----\n");
441         printf("!!! VOCÊ DIGITOU UM VALOR DIFERENTE DE (S/N) !!!\n");
442         printf("-----\n\n");
443     }
444
445     printf("digitou corretamente?(S/N)");
446     resp=getche();//capturar resposta
447     resp=toupper(resp);//forcar caracter para maiusculo
448     printf("\n\n");

```

```

449
450     if(resp!='S' && resp!='N'){//se o valor for diferente de S e N entao repete
451         flag=1;
452     }else{//senao para o loop
453         flag=0;
454     }
455 }
456 }while(flag==1);//repete enquanto o valor de resp for diferente de S e N
457 if(resp=='N'){//se nao estiver correta a data repete o procedimento
458     vdata(voo,aux);//chama a funcao data novamente se for respondido N
459 }
460
461
462 }
463
464 void datatostring(Tdados voo[TAM],Tdados *aux){//converte os campos de data inteiro e os joga para uma string
465     vdata(voo,aux);//chama funcao de ler data
466     char data[11],dia[3],mes[3],ano[5];
467     itoa(aux->dia, dia, 10);// converte inteiro dia para string dia
468     itoa(aux->mes, mes, 10);// converte inteiro mes para string mes
469     itoa(aux->ano, ano, 10);// converte inteiro ano para string ano
470     if(aux->dia<10){//se o dia for menor que DEZ atribui um digito ZERO a primeira posicao da string
471         strcpy(aux->data,"0");//adiciona ZERO a primeira posicao da string data
472         strcat(aux->data,dia);//concatena(soma, junta) a string data com a string dia
473     }
474     else{//senao copia normalmente o dia na sequencia dia/mes/ano
475         strcpy(aux->data,dia);//copia o valor da string dia para a string data
476     }
477     strcat(aux->data,"/");//concatena(soma, junta) a string data um caracter '/'
478
479     if(aux->mes<10){//se o dia for menor que DEZ atribui um digito ZERO a primeira posicao da string
480         strcat(aux->data,"0");//concatena(soma, junta) a string data o caracter '0'
481         strcat(aux->data,mes);//concatena(soma, junta) a string data a string mes
482     }
483     else{//senao copia normalmente o mes na sequencia dia/mes/ano
484         strcat(aux->data,mes);//concatena(soma, junta) a string data a string mes
485     }
486     strcat(aux->data,"/");//concatena(soma, junta) a string data um caracter '/'
487
488     if(aux->ano<10){//se o dia for menor que DEZ atribui um digito ZERO a primeira posicao da string
489         strcat(aux->data,"00");//concatena(soma, junta) a string data os caracteres'00'
490         strcat(aux->data,ano);//concatena(soma, junta) a string data a string ano
491     }
492     else{
493         if(aux->ano<100){//se o dia for menor que DEZ atribui um digito ZERO a primeira posicao da string
494             strcat(aux->data,"00");//concatena(soma, junta) a string data os caracteres'00'
495             strcat(aux->data,ano);//concatena(soma, junta) a string data a string ano
496         }
497         else{
498             if(aux->ano<1000){//se o dia for menor que DEZ atribui um digito ZERO a primeira posicao da string
499                 strcpy(aux->data,"0");//concatena(soma, junta) a string data o caracter'0'
500                 strcat(aux->data,ano);
501             }
502             else{//senao copia normalmente o ano na sequencia dia/mes/ano
503                 strcat(aux->data,ano);//concatena(soma, junta) a string data a string ano
504             }
505         }
506     }
507
508 void atualdata(Tdados voo[TAM],Tdados *aux){//captura a data atual para o sistema
509     vdata(voo,aux);//chamando a leitura de data
510     aux->idia=aux->dia;//joga o dia digitado para o dia inicial do programa
511     aux->imes=aux->mes;//joga o mes digitado para o mes inicial do programa
512     aux->iano=aux->ano;//joga o ano digitado para o ano inicial do programa
513 }
514
515 Tdados inserirpass(Tdados voo[TAM],Tdados *aux){//funcao usada no metododecadastro para inserir passageiros
516     strcpy(aux->nome,validanome());//copia o retorno da funcao para variavel nome
517     aux->origem=validaorigem();//atribui o retorno da funcao para variavel origem
518     aux->destino=validadestino(aux->origem);//atribui o retorno da funcao para variavel destino
519     //strcpy(aux->data,validdata());//atribui o retorno da funcao para a variavel data
520     datatostring(voo,aux);//chama a funcao que converte os campos dia me e ano para uma unica string DD/MM/AAAA
521     return *aux;//retorno da funcao
522 }
523
524 Tdados buscalista(Tdados voo[TAM],Tdados *aux){//funcao usada em relatorio 3 e 4 para obter os dados principais de busca
525     aux->origem=validaorigem();//atribui o retorno da funcao para variavel origem
526     aux->destino=validadestino(aux->origem);//atribui o retorno da funcao para variavel destino
527     //strcpy(aux->data,validdata());//atribui o retorno da funcao para a variavel data
528     datatostring(voo,aux);//chama a funcao que converte os campos dia me e ano para uma unica string DD/MM/AAAA
529     return *aux;//retorno da funcao
530 }
531
532 char *excluirnome(char nome[50]);//funcao usada para excluir nome do passageiro
533     strcpy(nome,"");//atribui vazio para a variavel
534     return nome;//retorno da funcao
535 }
536
537 void zeravetoraux(char vaux[16][50]);//procedimento para zerar vaux
538     int i;
539     for(i=0;i<16;i++){//varre o vetor de string e deixa todos campos vazios
540         strcpy(vaux[i],"");//atribui vazio para a posicao do vetor
541     }
542 }
543
544 void copiavetor(char lista[16][50]);//procedimento usado para jogar primeiro da fila de espera para lista de voo
545     int i,j;//contadores
546     char vaux[16][50];//variavel string
547     zeravetoraux(vaux);//deixa vazio o vetor auxiliar antes de usar..
548     j=0;//inicializando contador
549     for(i=0;i<16;i++){//copia do vetor original para o auxiliar sem os vazios
550         if(strcmp(lista[i],"")!=0){//se sao diferente de vazio
551             strcpy(vaux[j],lista[i]);//copia o conteudo da posicao de lista para a posicao de vaux
552             j++;//contador
553         }
554     }
555     for(i=0;i<16;i++){//copia do vetor auxiliar para vetor original
556         strcpy(lista[i],vaux[i]);//copia o conteudo da posicao de vaux devolta para a posicao de lista
557     }
558 }
559
560 int getPosvoo(Tdados voo[TAM],Tdados *aux){//encontra a posicao do voo especifico ja cadastrado

```

```

561 int pos,i;
562 pos=-1;//atribuir uma posicao diferente das possiveis de um vetor para ter certeza que encontrou a posicao
563 i=0;//inicializa o contador
564
565 while(pos!=-1 && i<TAM){//enquanto a posicao continuar -1 e o contador i for menor que o limite do vetor faça ...
566     if(
567         voo[i].origem==aux->origem &&
568         voo[i].destino==aux->destino &&
569         (strcmp(voo[i].data,aux->data)==0)){//se encontrar em alguma posicao do vetor origem e destino e data igual a digitada ...
570             pos=i;//salva a posicao que foi encontrado os itens iguais ao digitado
571         }
572     }
573     else{//senao continua o loop varrendo o vetor
574         i++;//acrescenta +1 ao contador i
575     }
576 }
577 //caso nao encontrar origem e destino e data iguais permanecera -1 para variavel pos
578 return pos;//retorna o valor da funcao
579 }
580
581 int getPosLivrepass(char lista[16][50]){// encontra a posicao livre para armazenar o passageiro no voo
582 int pos,i;//guardador de posicao ,contador
583 pos=-1;//atribuir uma posicao diferente das possiveis de um vetor para ter certeza que encontrou a posicao
584 i=0;//inicializa o contador i
585
586 while(pos!=-1 && i<4){//enquanto a posicao continuar -1 e o contador i for menor ou igual a 4 faça ...
587     if(strlen(lista[i])==0){//se tiver alguma posicao do vetor que esteja vazia me diga
588         pos=i;//armazena a posicao que foi encontrado vazio
589     }
590     i++;//acrescenta mais 1 ao contador i
591 }
592 //caso nao encontrar origem e destino e data iguais permanecera -1 para variavel pos
593 return pos;//retorna o valor da funcao
594 }
595
596
597 int getPosLivrepass2(char lista[16][50]){// encontra a posicao livre para armazenar o passageiro na fila de espera
598 int pos,i;//guardador de posicao ,contador
599 pos=-1;//atribuir uma posicao diferente das possiveis de um vetor para ter certeza que encontrou a posicao
600 i=5;//inicializa o contador i
601
602 while(pos!=-1 && i<14){//enquanto a posicao continuar -1 e o contador i for menor ou igual a 14 faça ...
603     if(strlen(lista[i])==0){//se tiver alguma posicao do vetor que esteja vazia me diga
604         pos=i;//armazena a posicao que foi encontrado vazio
605     }
606     i++;//acrescenta mais 1 ao contador i
607 }
608 //caso nao encontrar origem e destino e data iguais permanecera -1 para variavel pos
609 return pos;//retorna o valor da funcao
610 }
611
612
613 int getPosExcluido(char lista[16][50],Tdados *aux){//encontra a posicao que se encontra o nome do passageiro digitado num num
614     voo ou fila de espera
615     int pos,i;//guardador de posicao ,contador
616     pos=-1;//atribuir uma posicao diferente das possiveis de um vetor para ter certeza que encontrou a posicao
617     i=0;//inicializa o contador i
618
619     while(pos!=-1 && i<14){//enquanto a posicao continuar -1 e o contador i for menor ou igual a 14 faça ...
620         if(strcmp(lista[i],aux->nome)==0){//compara o nome digitado com os nomes armazenados na lista de voo e lista de espera
621             pos=i;//o primeiro que encontrar com este nome me diga
622         }
623         i++;//acrescenta +1 ao conta i
624     }
625     //caso nao encontrar origem e destino e data iguais permanecera -1 para variavel pos
626     return pos;// retorna a posicao a funcao
627 }
628
629 int getPosLivrevoo(Tdados voo[TAM],Tdados *aux){//encontra uma posicao no vetor para armazenar voo
630 int pos,i;//guardador de posicao ,contador
631 pos=-1;//atribuir uma posicao diferente das possiveis de um vetor para ter certeza que encontrou a posicao
632 i=0;//inicializa o contador i
633
634 while(pos!=-1 && i<TAM){//enquanto a posicao continuar -1 e o contador i for menor que o limite do vetor faça ...
635     if(
636         voo[i].origem==0 &&
637         voo[i].destino==0 &&
638         (strcmp(voo[i].data,"")==0) ){//se encontrar em alguma posicao do vetor em que origem e destino sejam 0 e data seja vazio
639         me diga ...
640         pos=i;//salva a posicao que foi encontrado os itens iguais ao digitado
641     }
642     else{//senao continua o loop varrendo o vetor
643         i++;//acrescenta +1 ao contador i
644     }
645 }
646 //caso nao encontrar origem e destino e data iguais permanecera -1 para variavel pos
647 return pos;//retorna o valor da funcao
648 }
649
650 int getnumeropassvoo(char lista[16][50],Tdados *aux){//conta quantos passageiros tem na lista de voo
651 int i;//contador i
652 aux->contvoo=0;//inicializando contador
653
654 for(i=0;i<4;i++){//varre a lista de voo e conta quantos nomes tem escrito
655     if(strlen(lista[i])!=0){
656         aux->contvoo++;//incrementa +1 ao contador de voo
657     }
658 }
659 //caso nao encontrar o contador permanece 0
660 return aux->contvoo;//retorna valor da funcao
661 }
662
663 int getnumeropassvooespera(char lista[16][50],Tdados *aux){//conta quantos passageiros tem na lista de espera
664 int i;//contador i
665 aux->contespera=0;//inicializando contador
666
667 for(i=5;i<14;i++){//varre a lista de espera e conta quantos nomes tem escrito
668     if(strlen(lista[i])!=0){
669         aux->contespera++;//incrementa +1 ao contador de voo
670     }
671 }

```

```

671 }
672 //caso nao encontrar o contador permanece 0
673 return aux->contespera;//retorna valor da funcao
674 }
675
676 void msgsucesso(){//procedimento de mensagem de reserva de voo realizado com sucesso
677     espacos();//insere enter na na tela
678     printf("-----\n");
679     printf(" Reserva de voo Realizada com Sucesso !!!\n");
680     printf("-----\n");
681     espacos();//insere enter na na tela
682     system("pause");//pausa a tela para o usuario ver a mensagem
683 }
684
685 void msgfiladeespera(){//procedimento de mensagem lista de voo lotada , voce foi colocado na fila de espera
686     espacos();//insere enter na na tela
687     printf("-----\n");
688     printf(" Lista de vôo Lotada !!!\n");
689     printf(" Você foi colocado na Fila de Espera.\n");
690     printf("-----\n");
691     espacos();//insere enter na na tela
692     system("pause");//pausa a tela para o usuario ver a mensagem
693 }
694
695 Tdados metodocadastro(Tdados voo[TAM],Tdados *aux){//funcao que controla o agendamento de voo
696     system("cls");//limpa tela
697     int posvoocad,posvoo,pospass;
698     inserirpass(voo,aux);// captura nome, destino , origem e data
699     posvoocad=getPosvoo(voo,aux);//procura se o voo ja esta cadastrado e retorna -1 senao tiver
700
701     if(posvoocad!=-1){//se o voo ta cadastrado
702         pospass=getPosLivrepass(voo[posvoocad].lista);//procura posicao livre na lista de voo para atribuir o passageiro
703
704         if(pospass!=-1){//se a posicao livre foi encontrada
705             strcpy(voo[posvoocad].lista[pospass],aux->nome);//adiciona o passageiro a lista de voo
706             msgsucesso();//exibir uma mensagem de sucesso no cadastro
707         }
708         else{//se nao encontro passageiro na lista de voo procura possicao livre na fila de espera
709             pospass=getPosLivrepass2(voo[posvoocad].lista);// captura a posicao livre na fila de espera para armazenar o nome
710             if(pospass!=-1){// se a posicao livre para adicionar o passageiro foi encontrada entao ...
711                 msgfiladeespera();//exibir uma mensagem que foi jogado para fila de espera
712                 strcpy(voo[posvoocad].lista[pospass],aux->nome);//atribui o nome do passageiro a fila de espera na posicao encontrada..
713             }
714             else{//senao foi entrado nenhuma possicao nem na lista de voo nem na fila de espera entao ...
715                 espacos();// adicionar enter na tela
716                 printf("Lista de Vôo e Lista de Espera Lotadas !!!\n");
717                 printf("Favor escolher outra data.\n");
718                 system("pause");//pausa a tela para o usuario ver a mensagem
719             }
720         }
721         voo[posvoocad].contvoo=getnumeropassvoo(voo[posvoocad].lista,aux);//obtem o numero atual de fila de voo
722         voo[posvoocad].contespera=getnumeropassvooespera(voo[posvoocad].lista,aux);//obtem o numero atual de fila de espera
723     }
724     else{// se o voo nao esta cadastrado entao ....
725         posvoo=getPosLivre(voo,aux);// procurando a possicao de voo livre , se nao encontrar posicao retorna -1
726         if(posvoo!=-1){//se encontrou uma posicao livre de voo para registrar entao ...
727             voo[posvoo].origem=aux->origem;//registra nessa possicao a origem
728             voo[posvoo].destino=aux->destino;//registra nessa possicao o destino
729             strcpy(voo[posvoo].data,aux->data);//registra nessa possicao a data
730             voo[posvoo].ano=aux->ano;//registra nessa possicao o ano
731             voo[posvoo].mes=aux->mes;//registra nessa possicao o mes
732             voo[posvoo].dia=aux->dia;//registra nessa possicao o dia
733             strcpy(voo[posvoo].lista[0],aux->nome);//registra nessa possicao o nome
734
735             voo[posvoo].contvoo=getnumeropassvoo(voo[posvoo].lista,aux);//obtem o numero atual de fila de voo
736             voo[posvoo].contespera=getnumeropassvooespera(voo[posvoo].lista,aux);//obtem o numero atual de fila de espera
737             msgsucesso();//exibir uma mensagem de sucesso no cadastro
738         }
739         else{//se nao encontrar possicao de voo é porque acabou o limite de registro para cadastros
740             espacos();//adiciona enter na tela
741             printf("Impossivel Cadastrar mais Vôos ao Programa os registros estao Lotados !!! \n");
742             system("pause");// pausa a tela para o usuario ver a mensagem
743         }
744     }
745
746     return *aux;//retorna o valor da funcao
747 }
748
749 void cabecalhorelatorio(char palavra[20],Tdados *aux){//procedimento de cabecalho para o relatorio 3 e 4
750
751     printf(" Lista %s \n",palavra);
752     printf("-----\n");
753     printf(" Origem .. : %s\n",v[aux->origem]);
754     printf(" Destino .. : %s\n",v[aux->destino]);
755     printf(" Data .... : %s\n",aux->data);
756     printf("-----\n");
757 }
758
759 void cabecalhorelatorio1(char palavra[20],Tdados voo[TAM],Tdados *aux,int total){//procedimento de cabecalho para o relatorio 1
760     int i;
761     if(total==--1){// nao axou possicao livre com getposlivrevoo é sinal de que todas todos registros estao lotados entao
762         imprimi tudo...
763         total=TAM;
764     }
765
766     printf(" Lista de %s\n",palavra);
767     printf("-----\n");
768     for(i=0;i<total;i++){//imprime todos voos cadastrados no sistema...
769         printf(" Origem ..... : %s\n",v[voo[i].origem]);
770         printf(" Destino ..... : %s\n",v[voo[i].destino]);
771         printf(" Data ..... : %s\n",voo[i].data);
772         printf(" N° de passageiros ..... : %d\n",voo[i].contvoo);
773         printf(" N° de passageiros na fila de espera .. : %d\n",voo[i].contespera);
774         printf("-----\n");
775     }
776     if(i==0){//se o indice i do for continuar zero é por que nao tem nenhum voo criados ainda..
777         espacos();//insere enter na tela
778         printf("-----\n");
779         printf(" NO MOMENTO NÃO EXISTEM VÔOS CRIADOS !!! \n");
780         printf("-----\n");
781         espacos();//insere enter na tela
782     }

```

```

782 }
783 }
784
785 void cabecalhorelatorio2(char palavra[20], Tdados voo[TAM], Tdados *aux, int total) { //procedimento de cabecalho para o relatorio 2
786 int i, flag; //variaveis
787 flag=0; //atribuindo zero a flag
788 if (total==1) { // nao axou posicao livre com getposslivrevoos é sinal de que todas todos registros estao lotados entao
    imprime tudo...
789     total=TAM;
790 }
791
792 printf(" Lista de %s\n", palavra);
793 printf("-----\n");
794 for(i=0; i<total; i++) { //imprime todos voo cadastrados porem ...
795     if(voo[i].contvoo>=2) { //..so imprime se for MAIOR ou IGUAL a 2
796         printf(" Origem .....: %s\n", voo[i].origem);
797         printf(" Destino .....: %s\n", voo[i].destino);
798         printf(" Data .....: %s\n", voo[i].data);
799         printf(" Nº de passageiros .....: %d\n", voo[i].contvoo);
800         printf(" Nº de passageiros na fila de espera ..: %d\n", voo[i].contespera);
801         printf("-----\n");
802         flag=1; //atribui 1 a variavel para evitar que imprima o proximo if
803     }
804 }
805
806 if(flag==0) { //se o indice i do for continuar zero é por que nao tem nenhum voo criados ainda..
807     espacos(); //inserir enter na tela
808     printf("-----\n");
809     printf(" NO MOMENTO NÃO EXISTEM VÔOS CONFIRMADOS !!! \n");
810     printf("-----\n");
811     espacos(); //inserir enter na tela
812 }
813 }
814
815 void exibe_r1(Tdados voo[TAM], Tdados *aux) { //procedimento de exibicao de relatorio 1
816 system("cls"); //limpa a tela
817 int total; //variavel de limite para usar na no procedimento de cabecalho
818 total=getPosLivrevoos(voo, aux); //captura uma posicao livre de voo
819 cabecalhorelatorio1("Vôos Criados", voo, aux, total); //chama o procedimento de relatorio enviando total por parametro
820
821 printf("\n\n\n\n\n\n\n\n\n\n"); //imprime alguns enter na tela
822 }
823
824 void exibe_r2(Tdados voo[TAM], Tdados *aux) { //procedimento de exibicao de relatorio 2
825 system("cls"); //limpa tela
826 int total; //variavel de limite para usar na no procedimento de cabecalho
827 total=getPosLivrevoos(voo, aux); //captura uma posicao livre de voo
828 cabecalhorelatorio2("Vôos Confirmados", voo, aux, total); //chama o procedimento de relatorio enviando total por parametro
829
830 printf("\n\n\n\n\n\n\n\n\n\n"); //imprime alguns enter na tela
831 }
832
833 void exibe_r3(char var[16][50], Tdados *aux) { //sub procedimento de exibicao do relatorio 3
834     system("cls"); //limpa tela
835     int i, li, lf, flag;
836     flag=0;
837     li=0; lf=4; //atribuindo limite inicial e final do for
838
839     cabecalhorelatorio("de Vôo", aux); //exibe o cabecalho do relatorio de 3
840
841     for(i=li; i<lf; i++) { //varre a lista de voo de zero a 4
842         if(!strcmp(var[i], "")==0) { //se for diferente de vazio imprime
843             printf("%dº. %s \n", i+1, var[i]); //deve se colocar +1 no i pois o vetor começa em zero
844             flag=1; //se a condicao de if passar apenas uma vez o flag sinaliza 1
845         }
846     }
847
848     if(flag==0) { //se nao for encontrado nenhum vazio no bloco for anterior flag continua ZERO entao imprime ....
849         printf("-----\n");
850         printf(" VÔO SEM PASSAGEIROS !!!\n");
851         printf("-----\n");
852     }
853
854     printf("\n\n\n\n\n\n\n\n\n\n"); // inserir enter na tela
855 }
856
857 void exibe_r3_lista(Tdados voo[TAM], Tdados *aux) { //procedimento de exibicao do relatorio 3
858     int indice;
859
860     buscaLista(voo, aux); //captura nome, origem, destino e data
861     indice=getPosvoo(voo, aux); //recebe da funcao a posicao do registro que foi encontrado o voo pedido
862     if(indice!=-1) { // se encontra o voo ...
863         exibe_r3(voo[indice].lista, aux); //exibe a lista de voo
864     }
865
866     else { //se nao encontrar exibe a mensagem ...
867         printf("-----\n");
868         printf(" VÔO NÃO CADASTRADO !!!\n");
869         printf("-----\n");
870     }
871 }
872 }
873
874 void exibe_r4(char var[16][50], Tdados *aux) { //sub procedimento de exibicao do relatorio 4
875     system("cls"); //limpa a tela
876     int i, li, lf, flag;
877     flag=0;
878     li=5; lf=14; //atribui o limite inicial e final para os controladores do for
879
880     cabecalhorelatorio("da fila de Espera", aux); //exibe o relatorio do cabecalho 3
881
882     for(i=li; i<lf; i++) { //varre a lista de espera de 5 a 14
883         if(!strcmp(var[i], "")==0) { //imprime todos nomes que nao estiverem vazios
884             printf("%dº. %s \n", i-5+1, var[i]);
885             flag=1; //se encontrar vazio atribui 1 a flag
886         }
887     }
888
889     if(flag==0) { //se nao foi encontrado nenhum nome no bloco anterior do for entao exibe
890         printf("-----\n");
891         printf(" FILA DE ESPERA VAZIA !!!\n");
892     }

```



```

893     printf("-----\n");
894 }
895
896     printf("\n\n\n\n\n\n\n\n\n\n"); //inserir enter na tela
897 }
898
899 void exibe_r4_lista(Tdados voo[TAM], Tdados *aux) { //procedimento de exibicao de relatorio 4
900     int indice;
901
902     buscalista(voo, aux); //captura nome , origem, destino , data
903     indice = getPosvoo(voo, aux); //atribui a variavel indice a posicao do vetor para o voo requisitado
904     if (indice != -1) { //se o voo for encontrado
905         exibe_r4(voo[indice].lista, aux); //exibe a lista de espera do voo especificado
906     }
907     else { //se nao for encontrado exibe a mensagem ...
908         printf("-----\n");
909         printf(" VÔO NÃO CADASTRADO !!!\n");
910         printf("-----\n");
911     }
912 }
913
914 void metodoexclusao(Tdados voo[TAM], Tdados *aux) { //procedimento de exclusao
915     char flag[15];
916     int posvoocad, pe; //posicao voo e posicao exclusao
917
918     inserirpass(voo, aux); //capturar nome, origem , destino , data
919     posvoocad = getPosvoo(voo, aux); //encontrar a posicao do registro que encontra esse passageiro
920     pe = getPosExcluido(voo[posvoocad].lista, aux); //encontrar a posicao da lista que encontra esse passageiro
921
922     if (pe != -1) { //encontrou a posicao na lista de passageiros
923         excluirnome(voo[posvoocad].lista[pe]); //excluir o passageiro
924         copiarvetor(voo[posvoocad].lista); //copia a lista original para um segundo vetor ignorando campos vazios e devolve a lista
925         original por referencia
926
927         if (pe >= 0 && pe <= 4) { strcpy(flag, "fila de voo"); } //se a posicao que foi encontrada estiver entre 0 e 4 entao foi excluido
928         do voo
929         else { strcpy(flag, "Fila de Espera"); } //se a posicao que foi encontrada estiver entre 5 e 14 entao foi excluido da fila de
930         espera
931
932         voo[posvoocad].contvoo = getnumeropassvoo(voo[posvoocad].lista, aux); //obtem o numero de passageiros atual de fila de voo
933         voo[posvoocad].contespera = getnumeropassvooespera(voo[posvoocad].lista, aux); //obtem o numero de passageiros atual de fila de
934         espera
935         espacos(); //inserir enter na tela
936         printf("-----\n");
937         printf("Passageiro encontrado na %s foi Excluido com Sucesso !!!\n", flag);
938         printf("-----\n");
939         espacos(); //inserir enter na tela
940         system("pause"); //pausar a tela para permitir que o usuario leia
941     } else { //se nao encontrou o nome do passageiro fornecido entao diga ...
942         espacos(); //inserir enter na tela
943         printf("-----\n");
944         printf("Não foi possível cancelar pois ,\n");
945         printf("o usuario informado não foi encontrado nessa origem ,destino e data !!!\n");
946         printf("-----\n");
947         espacos(); //inserir enter na tela
948         system("pause"); //pausar a tela para permitir que o usuario leia
949     }
950 }
951
952 void Exibedatahoje(Tdados *aux) { //procedimento para exibir a data do sistema
953     char v1[2], v2[2], v3[4]; //variaveis ajudantes
954     system("cls"); //limpa a tela
955     espacos(); //inserir enter na tela
956
957     if (aux->idia < 10) { strcpy(v1, "0"); } //se o dia for menor que 10 entao copia 0 para a variavel ajudante
958     else { strcpy(v1, ""); } //se nao deixa vazio
959
960     if (aux->imes < 10) { strcpy(v2, "0"); } //se o mes for menor que 10 entao copia 0 para a variavel ajudante
961     else { strcpy(v2, ""); } //se nao deixa vazio
962
963     if (aux->iano < 10) { strcpy(v3, "00"); } //se o ano for menor que 10 entao copia 00 para a variavel ajudante
964     else {
965         if (aux->iano < 100) { strcpy(v3, "00"); } //se o ano for menor que 100 entao copia 00 para a variavel ajudante
966         else {
967             if (aux->iano < 1000) { strcpy(v3, "0"); } //se o ano for menor que 1000 entao copia 0 para a variavel ajudante
968             else { strcpy(v3, ""); } //se nao deixa vazio
969         }
970     }
971
972     //utiliza-se a variaveis ajudante na frente dos valores reais para autocompletar com ZERO quando for necessario
973     printf("-----\n");
974     printf(" DATA DO SISTEMA ..: ");
975     printf(" %s%d/%s%d/%s%d \n", v1, aux->idia, v2, aux->imes, v3, aux->iano);
976     printf("-----\n");
977 }
978
979 void logotipo(Tdados voo[TAM], Tdados *aux) { // logotipo
980     Exibedatahoje(aux); //exibe a data do sistema
981     espacos(); //pular linhas
982     printf("-----\n");
983     printf("----- SEJA BEM VINDO -----\n");
984     printf("----- A -----\n");
985     printf("----- VOE FÁCIL LINHAS AÉREAS INTELIGENTES -----\n");
986     printf("-----\n\n");
987 }
988
989 int pergunta() { //funcao de pergunta do menu principal
990     int select;
991     printf("Escolha uma das opções acima: ");
992     scanf("%d", &select);
993     fflush(stdin); //limpar buffer de entrada
994     espacos(); //inserir enter na tela
995     return select; //retorno da funcao
996 }
997
998 int m1(Tdados voo[TAM], Tdados *aux) { //FUNCAO DE EXIBICAO DO MENU PRINCIPAL
999     int select;
1000     system("cls"); //limpar tela
1001     logotipo(voo, aux); //mostra logotipo
1002     printf("-----\n");
1003     printf("----- MENU PRINCIPAL ----- \n");

```

```

1001     printf("-----\n");
1002     printf("1 - Agendar vôo. \n");
1003     printf("2 - Cancelar passageiro. \n");
1004     printf("3 - Lista de relatórios. \n");
1005     printf("0 - Sair. \n");
1006     printf("-----\n");
1007     select=pergunta();//atribui o retor da funcao a variavel
1008     return select;//retorna o valor da funcao
1009 }
1010
1011 int m2 (){//FUNCAO DE EXIBICAO DO MENU RELATORIOS
1012     int select;
1013     espacos();//inser enter na tela
1014     espacos();//inser enter na tela
1015     printf("-----\n");
1016     printf("----- LISTA DE RELATORIOS ----- \n");
1017     printf("-----\n");
1018     printf("1 - Lista de vôos criados. \n");
1019     printf("2 - Lista de vôos Confirmados. \n");
1020     printf("3 - Lista dos passageiros de um determinado vôo. \n");
1021     printf("4 - Lista da espera de um determinado vôo. \n");
1022     printf("0 - Voltar ao menu anterior. \n");
1023     printf("-----\n");
1024     select=pergunta();//atribui a variavel o retorno da funcao chamada
1025     return select;//retorno da funcao
1026 }
1027
1028 int menu(Tdados voo[TAM],Tdados *aux){//funcao seletiva do menu principal
1029     int select;
1030
1031     do{
1032         select=m1(voo,aux);//atribui o retorno da funcao de menu para a variavel
1033     }while(!(select>=0 && select<=3));// repete enquanto nao estiver no intervalo de 0 a 3
1034
1035     switch (select){
1036         case 1:
1037             metodocadastro(voo,aux);// Chamando a lista de marcacao voo
1038             break;//interrompendo o switch
1039         case 2:
1040             metodoexclusao(voo,aux);// chamando a cancelamento passageiro
1041             break;//interrompendo o switch
1042         case 3:
1043             relatorios(voo,aux);// chamando a lista de relatorio
1044             break;//interrompendo o switch
1045         case 0:
1046             exit(0); // Fechar do programa (ignorando tudo)
1047             break;//interrompendo o switch
1048     }
1049     return select;//retorno da funcao
1050 }
1051
1052 void relatorios(Tdados voo[TAM],Tdados *aux){
1053     int select;
1054
1055     do{
1056         select=m2();//atribui o retorno da funcao
1057     }while(!(select>=0 && select<=4));// repete ateh que numero do relatorio desejado esteja no intervalo de 0 a 4
1058
1059     switch (select){
1060         case 1:
1061             exibe_r1(voo,aux);// Chamando a lista de Voos Criados
1062             system("pause");//pausar a tela para permitir que o usuario leia
1063             break;//interrompendo o switch
1064         case 2:
1065             exibe_r2(voo,aux);// Chamando a lista de Voos Confirmados
1066             system("pause");//pausar a tela para permitir que o usuario leia
1067             break;//interrompendo o switch
1068         case 3:
1069             exibe_r3_lista(voo,aux);// Chamando a lista de passageiros de um determinado Voo
1070             system("pause");//pausar a tela para permitir que o usuario leia
1071             break;//interrompendo o switch
1072         case 4:
1073             exibe_r4_lista(voo,aux);// Chamando a lista de Espera de um determinado Voo
1074             system("pause");//pausar a tela para permitir que o usuario leia
1075             break;
1076         case 0:
1077             menu(voo,aux); // voltar para menu principal
1078             break;//interrompendo o switch
1079     }
1080 }
1081 }
1082

```