```c
#include <windows.h>
#include <stdio.h>
#include <dos.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>


typedef struct
{
	float temp;
	char nome[20];
	struct elemento *prox;
}elemento;

elemento *F=NULL;
elemento *atual=NULL;
elemento *anterior=NULL;

elemento *alocarno()
{
	elemento *novo;
	novo=(elemento*)malloc(sizeof(elemento));
	novo->prox=NULL;
	return novo;
}

void cadastro()
{
	float temp;
	char nome[20];
	printf("\n\nCADASTRAR MOTOR\n NOME:");
	scanf("%s", &nome);
	printf(" TEMPERATURA MAXIMA DE OPERACAO:");
	scanf("%f", &temp);
	if(F==NULL)
	{
		F=alocarno();
		strcpy(F->nome,nome);
		F->temp=temp;
	}
	else
	{
		atual=F;
		while(atual->prox!=NULL)
		{
		 atual=(elemento*)atual->prox;
		}
		atual->prox=(struct elemento*)alocarno();
		atual=(elemento*)atual->prox;
		strcpy(atual->nome,nome);
		atual->temp=temp;
	}

	system("PAUSE");
}

int imprimir()
```

```
{
	int count = 0;

	if(F==NULL)
	{
		printf("Lista vazia");
		printf("\n");
	}
	else
	{
		count=1;
		atual=F;
		while(atual!=NULL)
		{
			printf("\n MOTOR %d\n Motor: %s \n",count, atual->nome);
			printf("Temperatura MAX OP: %f \n",atual->temp);
			atual=(elemento*)atual->prox;
			count++;
		}
	}

	return count;
}


float EscolherTemperatura()
{
	elemento *paux;
	int count=1, j=0, i=1;

	count = imprimir();

	if(count==0)
	{
		return 0;
	}
	else
	{
	printf("\nEscolha entre os motores cadastrados:");
	scanf("%d", &j);
	if(j>(count-1) || j<1)
	{
		printf("\n Numero invalido!!\n");
		return 0;
	}
	else
	{
		paux=F;
		while(paux->prox!=NULL)
			{
				if(j==i)
					break;
				else
					i++;
				paux=(elemento*)paux->prox;
			}

			return paux->temp;
```

```
	}
}
}

void limparlista()
{
	atual=F;
	while(atual!=NULL)
	{
		F=(elemento*)atual->prox;
		free(atual);
		atual=F;
	}

}

void excluir()
{

	int count = 1, i = 1, j = 0;

	count = imprimir();

	printf("\n Escolha motor para ser excluido: \n");
	scanf("%d", &j);

	if(j>(count-1) || j<1)
		printf("\n Numero invalido!!\n");
	else
	{
		anterior = NULL;
		atual = F;
		while(atual!=NULL)
			{

				if(i==j)
				{

					if(anterior==NULL)
					{
							F=(elemento*)atual->prox;
							free(atual);
							break;
					}
					else
					{
						anterior->prox=atual->prox;
						free(atual);
					}
					anterior->prox = atual->prox;
					free(atual);
					break;
				}
				else
				{
					i++;
					anterior = atual;
```

```
                                    atual = (elemento *) atual->prox;
                                }
                            }


        }


}

// Ler caractere
char SerialGetc(HANDLE *hCom)
{
   char rxchar;
   BOOL bReadRC;
   static DWORD iBytesRead;

   bReadRC = ReadFile(*hCom, &rxchar, 1, &iBytesRead, NULL);
   return rxchar;
}

// Escrever caractere
void SerialPutc(HANDLE hCom, char txchar)
{
   BOOL bWriteRC;
   static DWORD iBytesWritten;

   bWriteRC = WriteFile(hCom, &txchar, 1, &iBytesWritten,NULL);

   return;
}

// Ler string
char* SerialGets(HANDLE *hCom)
{
   static char rxstring[256];
   char c;
   int pos = 0;

   while(pos <= 255)
   {
      c = SerialGetc(hCom);
      if (c==13) break;
      if(c == '\r') continue; // discard carriage return
      rxstring[pos++] = c;
      if(c == '\n') break;

   }
   rxstring[pos] = 0;
   return rxstring;
}

// Escrever string
void SerialPuts(HANDLE *hCom, char *txstring)
{
   BOOL bWriteRC;
   static DWORD iBytesWritten;
   bWriteRC = WriteFile(*hCom, txstring, strlen(txstring), &iBytesWritten,NULL);
```

```
}

int main(int argc, char *argv[])
{
   FILE *p;
   DCB dcb;
   HANDLE hCom;
   BOOL fSuccess;
   LPCWSTR LpcCommPort = L"COM4";
   int i=0, opc,on=0;
   char t='1';
   float *vet;
   float *vet2;
   int tempo = 0;
   float temp;
   char temperatura[4],sensor[20];
   int loop = 0;
   int c;
   int cr;
   int vetor;

   hCom = CreateFile( LpcCommPort,
                   GENERIC_READ | GENERIC_WRITE,
                   0,    // must be opened with exclusive-access
                   NULL, // no security attributes
                   OPEN_EXISTING, // must use OPEN_EXISTING
                   0,    // not overlapped I/O
                   NULL  // hTemplate must be NULL for comm devices
                   );

   if (hCom == INVALID_HANDLE_VALUE)
   {
       // Handle the error.
       printf ("CreateFile failed with error %d.\n", GetLastError());
       return (1);
   }

   // Build on the current configuration, and skip setting the size
   // of the input and output buffers with SetupComm.

   fSuccess = GetCommState(hCom, &dcb);

   if (!fSuccess)
   {
      // Handle the error.
      printf ("GetCommState failed with error %d.\n", GetLastError());
      return (2);
   }

   // Fill in DCB: 57,600 bps, 8 data bits, no parity, and 1 stop bit.

   dcb.BaudRate = CBR_9600;     // set the baud rate
   dcb.ByteSize = 8;             // data size, xmit, and rcv
   dcb.Parity = NOPARITY;        // no parity bit
   dcb.StopBits = ONESTOPBIT;    // one stop bit

   fSuccess = SetCommState(hCom, &dcb);
```

```
   //

   if (!fSuccess)
   {
      // Handle the error.
      printf ("SetCommState failed with error %d.\n", GetLastError());
      return (3);
   }

   printf ("Serial port %s successfully reconfigured.\n", LpcCommPort);

   //SerialPuts(&hCom, "This is a text\n\nAnother line!\n");

   Sleep(2000);


   while(1)
   {
          system("cls");

      printf("OPCOES\n 1-CADASTRAR MOTOR\n 2-MOSTRAR SENSORES MOTOR\n 3-EXCLUIR
MOTOR\n 4-LIGAR MANUALMENTE(ON/OFF)\n 5-ESCOLHER MOTOR\n 6-SAIR\n ");
        scanf("%d", &opc);




              switch(opc)
                    {

                           case 1:
                           cadastro();
                           break;

                           case 2:
                           {
                                  vet=(float*)malloc(tempo*sizeof(float));
                                  vet2=(float*)malloc(tempo*sizeof(float));
                                  p=fopen("DadosMotor.txt","a");
                                 while(!loop)
                                        {

                                                printf("\MONITOR TEMPO REAL\n Aperte
E para sair.\n");
                                                tempo=1;
                                                 temp=6000;
                                                 sprintf(temperatura,"%.2f",temp);
                                                SerialPuts(&hCom, temperatura);
                                                 Sleep(1000);

                                                            while( i < tempo )
                                                            {
                                                                   if (
atof(SerialGets(&hCom)) == 0)  // alterna entre 0 e o valor, se esse é 0, o próximo
nao é;
                                                                   {
```

```
        vet[i]=atof(SerialGets(&hCom));

        fprintf(p,"Temperatura: %f \n",vet[i]);

                                                                }
                                                                Sleep(500);
                                                                if (
atof(SerialGets(&hCom)) == 0)  // alterna entre 0 e o valor, se esse é 0, o próximo
nao é;
                                                                {


        vet2[i]=atof(SerialGets(&hCom));

        fprintf(p,"Corrente: %f\n",vet2[i]);

        fprintf(p,"=====================================================\n",vet2[i]);
                                                                        i++;
                                                                }

                                                        }
                                                for ( i = 0; i < tempo; i++ )
                                                {
                                                        printf("\n Temperatura
%f ", vet[i]);
                                                        printf("\n Corrente %f
", vet2[i]);
                                                }
                                                i=0;
                                                Sleep(500);
                                                        system("cls");


                                        if ( kbhit())
                                        {
                                                c = getch();

                                                if ( c == 'E' || c=='e')
                                                {
                                                        fclose(p);
                                                        loop = 1;
                                                }
                                        }
                                }

                                break;
                        }
                        case 3:
                    excluir();
                        break;

                        case 4:
                                printf("\n Digite '1' para ON e '0' para OFF:
\n");
                                scanf("%i",&on);
                                if(on==0)
```

```
                                        {
                                               temp=-100;
                                               sprintf(temperatura,"%.2f",temp);
                                               SerialPuts(&hCom, temperatura);
                                        }else
                                               if(on==1)
                                               {
                                                      temp=100;
                                                      sprintf(temperatura,"%.2f",temp);
                                                      SerialPuts(&hCom, temperatura);
                                               }
                                         break;


                                  case 5:
                                         {
                                         temp = EscolherTemperatura();
                                        if ( temp!= 0 )
                                        {
                                               printf("\n Temperatura escolhida :
%f\n",temp);
                                               sprintf(temperatura,"%.2f",temp);
                                               SerialPuts(&hCom, temperatura);
                                        }
                                         }

                                         break;

                                  case 6:
                                         limparlista();
                                         exit(0);
                                        break;


                                  default:
                                          printf("Opcao nao existente");
                                 break;




                           }


      Sleep(1000);

   }
   SerialPuts(&hCom, "Fim!!\n\nMaravilha!!\n");

   SerialPutc(hCom,'1');
   printf ("Read value: %c\n", SerialGetc(&hCom));
   CloseHandle(hCom);
   system("PAUSE");
   return (0);
}
```