Programa E.41 *Gera a tabela T_r*

```
void GeraTr (TipoTr Tr)
{ int i, j, v, Soma = 0;
  for (i = 0; i <= MAXTRVALUE; i++)
  { Soma = 0;  v = i;
    for (j = 1; j <= 4; j++)
      { if ((v & 3) != NAOATRIBUIDO) Soma = Soma + 1;
        v = v >> 2;
      }
    Tr[i] = Soma;
  }
} /* GeraTr */
```

Programa E.42 *Função de transformação perfeita mínima*

```
TipoIndice hpm (TipoChave Chave,
                Tipor r,
                TipoTodosPesos Pesos,
                Tipog * g,
                TipoTr Tr,
                TipoK k,
                TipoTabRank *TabRank)
{ TipoIndice i, j, u, Rank, Byteg;
  u = hp (Chave, r, Pesos, g);
  j = u / k;        Rank = TabRank[j];
  i = j * k;        j  = i;
  Byteg = j / 4;  j = j + 4;
  while (j < u)
  { Rank = Rank + Tr[g[Byteg]];
    j = j + 4;  Byteg = Byteg + 1;
  }
  j = j - 4;
  while (j < u)
  { if (ObtemValor2Bits (g,j) != NAOATRIBUIDO) Rank = Rank+1;
    j = j + 1;
  }
  return Rank;
} /* hpm */
```

Programa E.43 *Procedimento para extrair palavras de um texto*

```
#define MAXALFABETO 255
#define TRUE  1
#define FALSE 0
typedef short  TipoAlfabeto[MAXALFABETO + 1];
FILE *ArqTxt, *ArqAlf;
TipoAlfabeto Alfabeto;
char Palavra[256];  char Linha[256];
int i;  short aux;
```


Continuação do Programa E.43

```
void DefineAlfabeto(short *Alfabeto)
{ char Simbolos[MAXALFABETO + 1];
  int i, CompSimbolos;
  char *TEMP;
  for (i = 0; i <= MAXALFABETO; i++)
    Alfabeto[i] = FALSE;
  fgets(Simbolos, MAXALFABETO + 1, ArqAlf);
  TEMP = strchr(Simbolos, '\n');
  if (TEMP != NULL) *TEMP = 0;
  CompSimbolos = strlen(Simbolos);
  for (i = 0; i < CompSimbolos; i++)
    Alfabeto[Simbolos[i]+127] = TRUE;
  Alfabeto[0] = FALSE;    /* caractere de codigo zero: separador */
}

int main(int argc, char *argv[])
{ ArqTxt = fopen(argv[1], "r");
  ArqAlf = fopen(argv[2], "r");
  DefineAlfabeto(Alfabeto);    /* Le alfabeto definido em arquivo */
  aux = FALSE;
  while (fgets(Linha, 256, ArqTxt) != NULL)
    { for (i = 1; i <= strlen(Linha); i++)
        { if (Alfabeto[Linha[i-1]+127])
            { sprintf(Palavra + strlen(Palavra), "%c", Linha[i-1]);
              aux = TRUE;
            }
          else
          if (aux)
            { puts(Palavra);
              *Palavra = '\0';
              aux = FALSE;
            }
        }
    }
  if (aux)
  { puts(Palavra);
    *Palavra = '\0';
  }
  fclose(ArqTxt);
  fclose(ArqAlf);
  return 0;
}
```