|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PREHĽAD ZÁKLADNÝCH LOGICKÝCH ŠTRUKTÚR ALGORITMOV V  C++** | | | | | |
| // **Priradenie dát**  P=1; | spojky  **Výstup dát** | | *// 1. Neúplný podmienený príkaz*  *if(P)*  *{*  *S;*  *}* | // 2.Úplný podmienený príkaz  if(P)  {  S1;  }  else  {  S2;  } | // 3. Prepínač  switch(K)  {  case k1:  S1;  break;  case k2:  S2;  break;  case ki:  Si;  break;  default:  Sj;  } |
| **Vstup dát** |
| //5.Cyklus s jednotkovým krokom zmeny  /spoužitím značky podmienky,  s použitím podmienky „if“ /    for(p=1;p<=n;p=p+1)  // pokiaľ je podmienka splnená,vykonáva sa S  {  S;  } | | | //11.Čítanie postupnosti  // vstupních dat {x}    int X[100];  int N; //rozsah 0-99  scanf("%d",&N);  for(int i=1; i<=N; i++)  {  scanf("%d", X[i]);  } | //5a.Cyklus s jednotkovým krokom zmeny  /spoužitím značky podmienky,  s použitím podmienky „if“ /  for(p=1;p<=n;p=p+1)  // pokiaľ je podmienka  splnená, vykonáva sa S  {  S;  } | //5b.Cyklus s jednotkovým krokom zmeny  /spoužitím značky podmienky, s použitím if/  P=1;  if(P>n)  // pokiaľ je podmienka p>n splnená  {  S1;  }  Else  // pokiaľ nie je podmienka p>n splnená,  {  S2;  p=p+1;  } |
| //6.Cyklus  // s podmienkou na začiatku    while(P)  {  S;  } | | // 7.Cyklus s podmienkou na konci  do  {  S;  }  while(P);  //S sa vykonáva //dovtedy, pokým //platí P | | // 4. Cyklus s výpočtom hodnôt  while(p==zoznam)  // pokiaľ je podmienka splnená,  // vykonáva sa S  {  S;  } |
| // 9.Program súčtu hodnôt prvkov  // postupnosti {z} vložený do premennej “S“  **#define n 11**  **// velkost pola musi byt konstanta, nie premenna**!  // velkost pola sa udava vzdy o 1 prvok vacsia!  int S=0;  int Z[n] ={ 1,2,3,4,5,6,7,8,9,10 };  for(int p=1; p<n; p++)  {  S = S + Z[p];  }  printf("%d",S); //Vysledok je S=54 | | | | //10.Výmena obsahu dvou premenných  **(PROGRAM k vyvoj.diagramu 10.b )**  int A=5, B=2, POM;  POM = A;  A = B;  B = POM;  printf("A=%d B=%d",A,B);  // Vysledok: A=2 B=5 |

|  |  |  |
| --- | --- | --- |
| // 12. Čítanie postupnosti  vstupních dat {z}  // s použitím cyklu s podmienkou na konci    double Z[100];  double KZ;  scanf("%f",&KZ);  int k=0;  do  {  k = k + 1;  scanf("%f",&Z[k]);  }  while(Z[k] != KZ);  int N = k - 1; | // 14. Výpočet maxima alebo minima zo zadaných hodnôt | |
| Výpočet maxima alebo minima  z 3 zadaných hodnôt  s použitím pomocnej premennej m  int A,B,C,M;  scanf (“%d %d %d“ , &A, &B, &C);  If(A>B)  M=A;  else  M=B;  if  (C>M)  M=C;  printf (“%d“,M); | Výpočet maxima alebo minima  z m - zadaných hodnôt postupnosti {z}  S použitím pomocnej premennej max    int m, max, z;  scanf (“%d %d“,&m, &max);  for (int i=2; i<=m; i++)  {  scanf(“%d“,8z);  if(z>max)  max=z;  }  printf(“%d“,max); |
| // 13. Čítanie postupnosti  vstupních dat {v}  // s použitím cyklu s podmienkou na začiatku  double V[100];  double KZ;  scanf("%f",&KZ);  int j=1;  scanf("%f",&V[j]);  while(V[j] != KZ)  {  j = j + 1;  scanf("%f",&V[j]);  }  int N = j - 1; | **JEDNODUCHÉ VYHĽADÁVANIE HODNOT**    int x,n, a[100],i;  scanf (“%d%d,&x,&n);  for (i=1;i<=n;i++)  scanf(“%d“,&a[i]);  i=0;  do  {  i=i+1;  }  while(x!=a[i] && i!=n);  if(a[i]==x)  printf(“PRVOK X JE I  V PORADI“);  else  printf(“SUBOR HLADANY PRVOK NEOBSAHUJE“); | **BINÁRNE VYHĽADÁVANIE HODNOT**  int x,n, a[100],s;  for (int i=1; i<=n; i++)  {  scanf(“%d“,&a[i]);  int d=1;  int h=n;  do  {  s=(d+h)/2;  if(a[s]<x)  d=s+1;  else  if(a[s]>x)  h=s-1;  }  While(x!=[s] && d>h);  If(d>h)  printf(“SUBOR HLADANY  PRVOK NEOBSAHUJE“);  else  printf(“PRVOK %d JE %d V PORADI“,x,s); |