Rešitve pisnega izpita z dne 30. junija 2003

1. naloga (25%)

```
void diamant(int n) {
  int i, j;

  for( i=0; i<n; i++ ) {
    for( j=n-i; j>0; j-- )
      printf(" ");
    for( j=0; j<(2*i+1); j++ )
      printf("*");
    printf("\n");
  }
  for( j=0; j<2*n+1; j++ )
    printf("*");
  printf("\n");
  for( i=n-1; i>=0; i-- ) {
    for( j=n-i; j>0; j-- )
      printf(" ");
    for( j=0; j<(2*i+1); j++ )
      printf("*");
    printf("\n");
  }
}
```

2. naloga (25%)

```
int tretji(int polje[], int vel) {
  int i,j,tmp;

  if( vel < 3 )
    return(-1);
  for( i=0; i<3; i++ )
    for( j=i+1; j<vel; j++ )
      if( polje[j] > polje[i] ) {
        tmp=polje[j];
        polje[j]=polje[i];
        polje[i]=tmp;
      }
  return(polje[2]);
}
```

3. naloga (25%)

```
#include <stdio.h>

main(int argc, char *argv[]) {
  FILE *fp1, *fp2;
  int ch1, ch2, st=1;

  fp1 = fopen(argv[1], "r");
  fp2 = fopen(argv[2], "r");
  while( ((ch1=getc(fp1)) != EOF) && ((ch2=getc(fp2)) != EOF) && (ch1 == ch2) ) {
    if( ch1 == '\n' )
      ++st;
  }
  if (ch1 == EOF)   // ce se je while zanka koncala, ker je konec prve datoteke
    ch2=getc(fp2);  // preberi se ustrezen znak v drugi datoteki
  if (ch1 != ch2)
    printf("Datoteki se razlikujeta v znaku %c in %c, v vrstici %d.\n", ch1, ch2, st);
  else
    printf("Datoteki sta enaki.\n");
  fclose(fp1);
  fclose(fp2);
}
```

4. naloga (25%)

```
struct element {
  int vred;
  struct element *nasl;
  struct element *pred;
};

// predpostavimo, da so elementi v vrsti urejeni NARASCAJOCE

void dodaj(struct element **p, int e) {
  struct element *q, *r;

  q = (struct element*) malloc(sizeof(struct element));
  q->vred = e;
  if ( (*p == NULL) || (e <= (*p)->vred) ) {  // dodaj na zacetek seznama
    q->nasl = *p;
    q->pred = NULL;
    if ( *p != NULL )
      (*p)->pred = q;
    *p = q;
  } else {
      // poisci ustrezno mesto v seznamu za nov element
      r = *p;
      while ( (r->nasl != NULL) && (r->nasl->vred < e) )
        r = r->nasl;
      // dodaj za element, na katerega kaze r
      q->nasl = r->nasl;
      if (r->nasl != NULL)
        r->nasl->pred = q;
      r->nasl = q;
      q->pred = r;
  }
}
```