Datoteke zaglavlja

Standardne datoteke zaglavlja

```
<assert.h>       <ctype.h>        <errno.h>
<float.h>        <limits.h>       <locale.h>
<math.h>         <setjmp.h>       <signal.h>
<stdarg.h>       <stddef.h>       <stdio.h>
<stdlib.h>       <string.h>       <time.h>
```

<math.h> - primjeri funkcija

- double atan(double x);
- double atan2(double y, double x);

```
int main( ){
   double x, y = -1.;

   printf("%g\n", atan(y/x));        /* 0.785398 */
   printf("%g\n", atan2(y, x));      /* -2.35619 */

   printf("%g\n", atan(1 < 5 < 3));

   return 0;
}
```

Funkcije `floor` i `ceil`

- `double floor(double x);`
- `double ceil(double x);`

Primjer:

```
printf("%g\n", floor(5.2));
printf("%g\n", floor(-5.2));

printf("%g\n", ceil(5.2));
printf("%g\n", ceil(-5.2));
```

Funkcija fmod

- `double fmod(double x, double y);`

Primjer:

```
printf("%g\n", fmod(5.2, 2.6));
printf("%g\n", fmod(5.57, 2.51));
printf("%g\n", fmod(5.57, -2.51));
printf("%g\n", fmod(-5.57, 2.51));
printf("%g\n", fmod(-5.57, -2.51));
```

Funkcija `frexp`

- `double frexp(double x, int *exp);`

Primjer:

```
double x = 8.0;
int exp;

printf("%f\n", frexp(x, &exp));
printf("%d\n", exp);
```

Funkcije exp, log, log10 i pow

- double exp(double x);
- double log(double x);
- double log10(double x);
- double pow(double x, double y);

Primjer:

```
printf("%g\n", log(exp(22.0)));
printf("%g\n", log10(pow(10.0, 22.0)));
```

<ctype.h> - primjeri funkcija

#include <ctype.h>

…

int c;

- int isalpha (c); /* 0 ili ≠ 0 */
- int isupper (c);
- int ispunct (c);
- int isspace (c);
- int isdigit (c);
- int isxdigit (c);
- int iscntrl (c);

Primjer:

- int tolower (c);
- int toupper (c);
- int toascii (c);
- …

```
for (i = 0; polje[i] != 0; i++)
   if (islower(polje[i]))
      polje[i] = toupper(polje[i]);
```

<stdlib> - primjeri funkcija

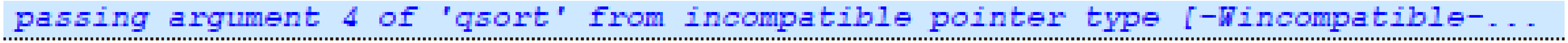
- void qsort(void *base, size_t n, size_t size,
  int (*comp)(const void *, const void *)) ;

Primjer:

```
int main(){
   char rjecnik[3][20] = {"po","ut","sr"};

   qsort(rjecnik, 3, 20, strcmp);
   puts(rjecnik[2]);
   return 0;
}
```

Upozorenje:



```
passing argument 4 of 'qsort' from incompatible pointer type [-Wincompatible-...
```

Funkcija bsearch

-
  ```
  void *bsearch(const void *key, const void *base,
                           size_t n, size_t size,
                       int (*comp)(const void *, const void *));
  ```

Primjer:

```
printf("%s", bsearch("ut", rjecnik, 3, 20, usporedi));

int usporedi(const void *a, const void *b)
{
   return (strcmp((char *)a, (char *)b));
}
```

Definiranje kriterija sortiranja

Primjer:

```
int main()
{
    int i, polje[4] = {1, 3, -4, 3};

    qsort(polje, 4, sizeof(int), usporedi);

    for(i = 0; i < 4; i++)
        printf("%d\n", polje[i]);
    return 0;
}
```

Funkcija usporedi

```
int usporedi(const void *a, const void *b)
{
    if (*(int*)a < *(int*)b)
        return -1;
    else if (*(int*)a > *(int*)b)
        return 1;
    else
        return 0;
}
```

ili samo

```
    return (*(int *)a - *(int*)b);
```

Funkcije rand i srand

- int rand(void);
- void srand(unsigned int seed);

Primjer:

```
int seed;

scanf("%d", &seed);
srand(seed); /* inicijalizacija */

printf("%d\n", rand());
printf("%d\n", rand());
printf("%d\n", RAND_MAX);        /* 32767 */
```