

Introduction to Computing

Jerzy Nawrocki
Faculty of Computing & Telecom.
Poznan University of Technology
jerzy.nawrocki@put.poznan.pl

Text processing

<https://ultimatehistoryproject.com/the-medieval-scribe.html>

Introduction to Computing

Introduction

We have already discussed:

- **Integer numbers,**


It's time for:

- **text.**

Text processing (2)

Introduction to Computing

Aim of the lecture




To present:

- **fundamentals of text processing,**
- **rule-based programming,**
- **regular expressions.**

Text processing (3)

Introduction to Computing




Text = Sequence of characters

Text processing (4)

Introduction to Computing

Agenda



- **Individual characters (C, Python)**
- **Strings of characters (C, Python)**
- **Rule-based text processing (AWK)**
- **Regular expressions (AWK, Python, C)**

<https://ultimatehistoryproject.com/the-medieval-scribe.html>

Text processing (5)

Introduction to Computing

For-loop as a shorthand notation

```
for (init, cond, next) {  
    body;  
}
```

```
init;  
while (cond) {  
    body;  
    next;  
}
```

C

Text processing (6)

Introduction to Computing

What will the result be?

3
1
2
3

C

```
#include <stdio.h>
int main(void) {
    int n, x, i, sum;
    scanf("%d", &n);
    sum= 0;
    i= 0;
    while (i < n){
        scanf("%d", &x);
        sum+= x;
        i+= 1; }
    printf("%d\n", sum);}
```

C

```
#include <stdio.h>
int main(void) {
    int n, x, i, sum;
    scanf("%d", &n);
    sum= 0;

    for(i= 0; i < n; i+= 1){
        scanf("%d", &x);
        sum+= x; }

    printf("%d\n", sum);}
```

6

Text processing (7)

Introduction to Computing

For-loop

for variable in sequence :
... Variable ...

for variable in range (first, uBound, step) :
... Variable ...

Text processing (8)

Introduction to Computing

What will the result be?

3
1
2
3

Python

```
n= int(input())
sum= 0
i= 0
while i < n:
    sum+= int(input())
    i+= 1
print(sum)
```

Python

```
n= int(input())
sum= 0

for i in range(0, n, 1):
    sum+= int(input())

print(sum)
```

6

Text processing (9)

Introduction to Computing

Shortcuts

for variable in range (first, uBound, 1) :
... Variable ...

≡

for variable in range (first, uBound) :
... Variable ...

≡

for variable in range (0, uBound) :
... Variable ...

≡

for variable in range (uBound) :
... Variable ...

Text processing (10)

Introduction to Computing

What will the result be?

3
1
2
3

What if it's not known

Python

```
n= int(input())
sum= 0
i= 0
while i < n:
    sum+= int(input())
    i+= 1
print(sum)
```

C

```
#include <stdio.h>
int main(void){
    int n, x, i, sum;
    scanf("%d", &n);
    sum= 0;
    i= 0;
    while (i < n){
        scanf("%d", &x);
        sum+= x;
        i+= 1; }
    printf("%d\n", sum);}
```

6

Text processing (11)

Introduction to Computing

Reading unknown number of data?

1 2
3

Python

```
import sys
sum= 0
for line in sys.stdin:
    for x in line.split():
        sum+= int(x)
print(sum)
```

6

Text processing (12)

Introduction to Computing

scanf is a function

Value of **scanf** = number of items read or EOF if end-of-file is reached

C

```
scanf("format", addresses_of_vars)
```

Function can be called as a procedure – then its value is neglected.

Text processing (13)

Introduction to Computing

Reading unknown number of data?

1 2
3

```
#include <stdio.h>
int main(void) {
    int x, sum;
    sum = 0;
    while (scanf("%d", &x) != EOF) {
        sum += x;
    }
    printf("%d\n", sum);
}
```

C

-1

6

Text processing (14)

Introduction to Computing

Initial values

type var₁ = value₁, ..., var_n = value_n ;

```
int x, y;
...
x = 0;
y = 1;
...
```

≡

```
int x = 0, y = 1;
...
```

C

Text processing (15)

Introduction to Computing

Initial values of arrays

type name [] = {value₁, ..., value_n};

```
int Val[3];
...
Val[0] = 2;
Val[1] = 3;
Val[2] = 4;
...
```

≡

```
int Val[] = {2, 3, 4};
...
```

C

Text processing (16)

Introduction to Computing

Initial values of arrays

type name [] = {value₁, ..., value_n};

```
int Val[3];
...
Val[0] = 2;
Val[1] = 3;
Val[2] = 4;
...
```

≡

```
int Val[] = {2, 3, 4};
...
```

C

Text processing (17)

Introduction to Computing

Quiz

What will the result be?

```
#include <stdio.h>
int main(void) {
    int i, a[] = { 5, 3, 0, 7, 1, 1, 2, 2, 0 };
    for(i = 0; a[i] > 0; i = i + 1)
        ;
    printf("%d \n", i);
}
```

2

C

Text processing (18)

Introduction to Computing

Jerzy Nawrocki
Faculty of Computing & Telecom.
Poznan University of Technology
jerzy.nawrocki@put.poznan.pl

Text processing

<https://ultimatehistoryproject.com/the-medieval-scribe.html>

Introduction to Computing

Agenda


- Individual characters (C, Python)
- Strings of characters (C, Python)
- Rule-based text processing (AWK)
- Regular expressions (AWK, Python, C)

<https://ultimatehistoryproject.com/the-medieval-scribe.html>

Text processing (20)

Introduction to Computing

Text = Sequence of characters

 **C**

<https://ultimatehistoryproject.com/the-medieval-scribe.html>


Text processing (21)

Introduction to Computing

Single characters

Variable declaration

```
#include <stdio.h>
int main(void) {
    char c1;
    c1 = 'A';
    printf("%c\n", c1);
    return 0; }
```

 **C**

A


Text processing (22)

Introduction to Computing

Single characters

Assignment of a constant value

```
#include <stdio.h>
int main(void) {
    char c1;
    c1 = 'A';
    printf("%c\n", c1);
    return 0; }
```

 **C**

A


Text processing (23)

Introduction to Computing

Single characters

Format specifier

```
#include <stdio.h>
int main(void) {
    char c1;
    c1 = 'A';
    printf("%c\n", c1);
    return 0; }
```

 **C**

A

Text processing (24)

Introduction to Computing

Single characters

```
#include <stdio.h>
int main(void) {
    char c1;
    c1= 'A';
    printf("%c\n", c1);
    return 0; }
```

A

```
#include <stdio.h>
int main(void) {
    char c1= 'A';
    printf("%c\n", c1);
    return 0; }
```

Text processing (25)

Introduction to Computing

Single characters

Hello world!

```
c1= input()
print(c1[0])
```


H

```
#include <stdio.h>
int main(void) {
    char c1;
    scanf("%c", &c1);
    printf("%c\n", c1);
    return 0; }
```

Text processing (26)

Introduction to Computing

Character vs. text



'A' - text
"A" - text

C

'A' - single char
"A" - text

Text = Sequence of characters

Text processing (27)

Introduction to Computing

Quiz: What will the result be?

Begin with the end in mind
(-) Stephen Covey

```
#include <stdio.h>
int main(){
    char ch;
    while (scanf("%c", &ch) != EOF)
        ;
    printf("%c", ch); }
```

C

Text processing (28)

Introduction to Computing

ASCII: American Standard Code for Information Interchange

DEL	BEL	BS	HT	LF	VT	FF	CR	SO	SI
0000	0001	0002	0003	0004	0005	0006	0007	0008	0009
0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99
100	101	102	103	104	105	106	107	108	109
110	111	112	113	114	115	116	117	118	119
120	121	122	123	124	125	126	127	128	129

Text processing (29)


Introduction to Computing

Type conversion

A value of one type → Corresponding value of another type

Explicit:

- Special function



Integer

Text

```
X = int(input())
print(X + 1)
```

Text processing (30)

Introduction to Computing

Type conversion

A value of one type → Corresponding value of another type

Explicit:

- Special function
- **Type casting**

C

```
int t;
char last;

last= (char) t;
```

New type

Text processing (31)

Introduction to Computing

Type conversion

A value of one type → Corresponding value of another type

Explicit:

- Special function
- **Type casting**

Implicit: type coercion

C

```
int t;
char last;

last= t;
```

New type is inferred

Text processing (32)

Introduction to Computing

Other input-output subroutines

C

```
int getchar(void)
    Reads one character from the standard input.
    Returns EOF if the input is empty.

int putchar(int)
    Writes one character to the standard output.
    Returns the written character or EOF in case of error.
```

Text processing (33)

Introduction to Computing

Type coercion (automatic conversion)

Begin with the end in mind
(-) Stephen Covey

C

```
#include <stdio.h>
int main(){
    int t;
    char last;
    while ((t = getchar()) != EOF)
        last= t;
    putchar(last);
    putchar('\n');
```

Type coercion

y

Text processing (34)

Introduction to Computing

What will the result be?

Python

```
import sys
for line in sys.stdin:
    for ch in line:
        last= ch
print(last)
```

y

Text processing (35)

Introduction to Computing

ord() and chr()

Python

```
code == ord(character)
character == chr(code)
```

Integer

Character

Character

Integer

Text processing (36)

© Jerzy Nawrocki, Introduction to Computing

Introduction to Computing									
ASCII: American Standard Code for Information Interchange									
DEL	SP	SP	SP	SP	SP	SP	SP	SP	SP
0000	0001	0002	0003	0004	0005	0006	0007	0008	0009
0	1	2	3	4	5	6	7	8	9
16	17	18	19	20	21	22	23	24	25
26	27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63	64	65
66	67	68	69	70	71	72	73	74	75
76	77	78	79	80	81	82	83	84	85
86	87	88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103	104	105
106	107	108	109	110	111	112	113	114	115
116	117	118	119	120	121	122	123	124	125
126	127	128	129	130	131	132	133	134	135
136	137	138	139	140	141	142	143	144	145
146	147	148	149	150	151	152	153	154	155
156	157	158	159	160	161	162	163	164	165
166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185
186	187	188	189	190	191	192	193	194	195
196	197	198	199	200	201	202	203	204	205
206	207	208	209	210	211	212	213	214	215
216	217	218	219	220	221	222	223	224	225
226	227	228	229	230	231	232	233	234	235
236	237	238	239	240	241	242	243	244	245
246	247	248	249	250	251	252	253	254	255
256	257	258	259	260	261	262	263	264	265
266	267	268	269	270	271	272	273	274	275
276	277	278	279	280	281	282	283	284	285
286	287	288	289	290	291	292	293	294	295
296	297	298	299	300	301	302	303	304	305
306	307	308	309	310	311	312	313	314	315
316	317	318	319	320	321	322	323	324	325
326	327	328	329	330	331	332	333	334	335
336	337	338	339	340	341	342	343	344	345
346	347	348	349	350	351	352	353	354	355
356	357	358	359	360	361	362	363	364	365
366	367	368	369	370	371	372	373	374	375
376	377	378	379	380	381	382	383	384	385
386	387	388	389	390	391	392	393	394	395
396	397	398	399	400	401	402	403	404	405
406	407	408	409	410	411	412	413	414	415
416	417	418	419	420	421	422	423	424	425
426	427	428	429	430	431	432	433	434	435
436	437	438	439	440	441	442	443	444	445
446	447	448	449	450	451	452	453	454	455
456	457	458	459	460	461	462	463	464	465
466	467	468	469	470	471	472	473	474	475
476	477	478	479	480	481	482	483	484	485
486	487	488	489	490	491	492	493	494	495
496	497	498	499	500	501	502	503	504	505
506	507	508	509	510	511	512	513	514	515
516	517	518	519	520	521	522	523	524	525
526	527	528	529	530	531	532	533	534	535
536	537	538	539	540	541	542	543	544	545
546	547	548	549	550	551	552	553	554	555
556	557	558	559	560	561	562	563	564	565
566	567	568	569	570	571	572	573	574	575
576	577	578	579	580	581	582	583	584	585
586	587	588	589	590	591	592	593	594	595
596	597	598	599	600	601	602	603	604	605
606	607	608	609	610	611	612	613	614	615
616	617	618	619	620	621	622	623	624	625
626	627	628	629	630	631	632	633	634	635
636	637	638	639	640	641	642	643	644	645
646	647	648	649	650	651	652	653	654	655
656	657	658	659	660	661	662	663	664	665
666	667	668	669	670	671	672	673	674	675
676	677	678	679	680	681	682	683	684	685
686	687	688	689	690	691	692	693	694	695
696	697	698	699	700	701	702	703	704	705
706	707	708	709	710	711	712	713	714	715
716	717	718	719	720	721	722	723	724	725
726	727	728	729	730	731	732	733	734	735
736	737	738	739	740	741	742	743	744	745
746	747	748	749	750	751	752	753	754	755
756	757	758	759	760	761	762	763	764	765
766	767	768	769	770	771	772	773	774	775
776	777	778	779	780	781	782	783	784	785
786	787	788	789	790	791	792	793	794	795
796	797	798	799	800	801	802	803	804	805
806	807	808	809	810	811	812	813	814	815
816	817	818	819	820	821	822	823	824	825
826	827	828	829	830	831	832	833	834	835
836	837	838	839	840	841	842	843	844	845
846	847	848	849	850	851	852	853	854	855
856	857	858	859	860	861	862	863	864	865
866	867	868	869	870	871	872	873	874	875
876	877	878	879	880	881	882	883	884	885
886	887	888	889	890	891	892	893	894	895
896	897	898	899	900	901	902	903	904	905
906	907	908	909	910	911	912	913	914	915
916	917	918	919	920	921	922	923	924	925
926	927	928	929	930	931	932	933	934	935
936	937	938	939	940	941	942	943	944	945
946	947	948	949	950	951	952	953	954	955
956	957	958	959	960	961	962	963	964	965
966	967	968	969	970	971	972	973	974	975
976	977	978	979	980	981	982	983	984	985
986	987	988	989	990	991	992	993	994	995
996	997	998	999	1000	1001	1002	1003	1004	1005
1006	1007	1008	1009	1010	1011	1012	1013	1014	1015
1016	1017	1018	1019	1020	1021	1022	1023	1024	1025
1026	1027	1028	1029	1030	1031	1032	1033	1034	1035
1036	1037	1038	1039	1040	1041	1042	1043	1044	1045
1046	1047	1048	1049	1050	1051	1052	1053	1054	1055
1056	1057	1058	1059	1060	1061	1062	1063	1064	1065
1066	1067	1068	1069	1070	1071	1072	1073	1074	1075
1076	1077	1078	1079	1080	1081	1082	1083	1084	1085
1086	1087	1088	1089	1090	1091	1092	1093	1094	1095
1096	1097	1098	1099	1100	1101	1102	1103	1104	1105
1106	1107	1108	1109	1110	1111	1112	1113	1114	1115
1116	1117	1118	1119	1120	1121	1122	1123	1124	1125
1126	1127	1128	1129	1130	1131	1132	1133	1134	1135
1136	1137	1138	1139	1140	1141	1142	1143	1144	1145
1146	1147	1148	1149	1150	1151	1152	1153	1154	1155
1156	1157	1158	1159	1160	1161	1162	1163	1164	1165
1166	1167	1168	1169	1170	1171	1172	1173	1174	1175
1176	1177	1178	1179	1180	1181	1182	1183	1184	1185
1186	1187	1188	1189	1190	1191	1192	1193	1194	1195
1196	1197	1198	1199	1200	1201	1202	1203	1204	1205
1206	1207	1208	1209	1210	1211	1212	1213	1214	1215
1216	1217	1218	1219	1220	1221	1222	1223	1224	1225
1226	1227	1228	1229	1230	1231	1232	1233	1234	1235
1236	1237	1238	1239	1240	1241	1242	1243	1244	1245
1246	1247	1248	1249	1250	1251	1252	1253	1254	1255
1256	1257	1258	1259	1260	1261	1262	1263	1264	1265
1266	1267	1268	1269	1270	1271	1272	1273	1274	1275
1276	1277	1278	1279	1280	1281	1282	1283	1284	1285
1286	1287	1288	1289	1290	1291	1292	1293	1294	1295
1296	1297	1298	1299	1300	1301	1302	1303	1304	1305
1306	1307	1308	1309	1310	1311	1312	1313	1314	1315
1316	1317	1318	1319	1320	1321	1322	1323	1324	1325
1326	1327	1328	1329	1330	1331	1332	1333	1334	1335
1336	1337	1338	1339	1340	1341	1342	1343	1344	1345
1346	1347	1348	1349	1350	1351	1352	1353	1354	1355
1356	1357	1358	1359	1360	1361	1362	1363	1364	1365
1366	1367	1368	1369	1370	1371	1372	1373	1374	1375
1376	1377	1378	1379	1380	1381	1382	1383	1384	1385
1386	1387	1388	1389	1390	1391	1392	1393	1394	1

Introduction to Computing

Quiz: What will the result be?

000	001	002
16	17	18
32	33	34
48	49	50
64	65	66
80	81	82
96	97	98
112	113	114

```

x= 'B'
ch= ord(x) + (ord('a') - ord('A'))
print(chr(ch), end='')
    
```

32

Text processing (43)

Introduction to Computing

Quiz: What will the result be?

Ali Baba and the Forty Thieves

```

import sys
for line in sys.stdin:
    for x in line:
        if 'A' <= x and x <= 'Z':
            ch= ord(x) + ord('a') - ord('A')
            print(chr(ch), end='')
        else:
            print(x, end='')
    
```

Text processing (44)

Introduction to Computing

Quiz: What will the result be?

Ali Baba and the Forty Thieves

```

#include <stdio.h>
int main(){
    char x, ch;
    while (scanf("%c", &x) != EOF){
        if ('A' <= x && x <= 'Z'){
            ch= x + ('a' - 'A');
            printf("%c", ch);}
        else
            printf("%c", x);
    }
}
    
```

ali baba and the forty thieves

Text processing (45)

Introduction to Computing

Agenda

- Individual characters (C, Python)
- Strings of characters (C, Python)
- Rule-based text processing (AWK)
- Regular expressions (AWK, Python, C)

<https://ultimatehistoryproject.com/the-medieval-scribe.html>

Text processing (46)

Introduction to Computing

String of characters

I like PUT

```

char Text[]="I like PUT";
    
```

End marker

I		l	i	k	e		P	U	T	\0
0	1	2	3	4	5	6	7	8	9	10

length

```

Text= "I like PUT"
    
```

10	I		l	i	k	e		P	U	T
0	1	2	3	4	5	6	7	8	9	

Text processing (47)

Introduction to Computing

Numerical representation

ASCII

000	001	002
16	17	18
32	33	34
48	49	50
64	65	66
80	81	82
96	97	98
112	113	114

I like PUT

73	32	108	105	107	101	32	80	85	84	0
0	1	2	3	4	5	6	7	8	9	10

10	73	32	108	105	107	101	32	80	85	84
0	1	2	3	4	5	6	7	8	9	

Text processing (48)

Introduction to Computing

Reading / writing a text

C

```
char Text[]="I like PUT";


scanf("%s", Text);
printf("%s", Text);
```

Text processing (49)


Introduction to Computing

Single characters

Hello world!



```
msg= input()
print(msg)
```



```
#include <stdio.h>
int main(void) {
    char msg[100];
    scanf("%s", msg);
    printf("%s\n", msg);
    return 0; }
```

Hello world!

Hello

Text processing (50)

Introduction to Computing

Simple text processing

Hello world!



```
msg= input()
print(msg)
```




```
#include <stdio.h>
int main(void) {
    char msg[100];
    fgets(msg, 100, stdin);
    printf("%s\n", msg);
    return 0; }
```

Hello world!

Text processing (51)

Introduction to Computing

Length of text



```
Text= "Go"
print(len(Text))
```

2

Text processing (52)

Introduction to Computing

Other useful functions from **string.h**

C

```
int strlen(const char *src)
/* length of string pointed to by src */
```

Text processing (53)

Introduction to Computing

Quiz

C

```
#include <stdio.h>
#include <string.h>
int main() {
    char Text[]="Go";
    printf("%d\n", strlen(Text));
    printf("%d\n", sizeof Text);
}
```


2

3

Text processing (54)

Introduction to Computing

What will the result be?



Warsaw


```
Expected= "Warsaw"
Provided= input()
if Expected == Provided:
    print("OK")
else:
    print("Wrong!")
```

OK

Text processing (55)

Introduction to Computing

What will the result be?



Warsaw

```
#include <stdio.h>
int main(){
    char Expected[] = "Warsaw";
    char Provided[100];
    scanf("%s", &Provided);
    if (Expected == Provided)
        printf("OK\n");
    else printf("Wrong!\n"); }
```

Text processing (56)

Introduction to Computing



What?

```
1. #include <stdio.h>
2. int main(){
3.     char Expected[] = "Warsaw";
4.     char Provided[100];
5.     scanf("%s", &Provided);
6.     if (Expected == Provided)
7.         printf("OK\n");
8.     else printf("Wrong!\n");
9.     return 0; }
```


Success #stdin #stdout 0s 5460KB

stdin
Warsaw

stdout
Wrong!

Text processing (57)

What will the result be?



Warsaw


```
#include <string.h>
#include <stdio.h>
int main(){
    char Expected[] = "Warsaw";
    char Provided[100];
    scanf("%s", &Provided);
    if (!strcmp(Expected, Provided))
        printf("OK\n");
    else printf("Wrong!\n");
    return 0; }
```

OK

Text processing (58)

Introduction to Computing

Agenda

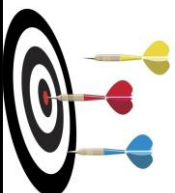


- Individual characters (C, Python)
- Strings of characters (C, Python)
- Rule-based text processing (AWK)
- Regular expressions (AWK, Python, C)

Text processing (59)

Introduction to Computing

Aim of the lecture




To present:

- Another programming paradigm: rule-based programming
- Rudiments of AWK

Text processing (60)

Introduction to Computing

Law of instrument




If all you have is a hammer, everything looks like a nail

(-) Abraham Maslow

Text processing (61)

Introduction to Computing

Origins of AWK



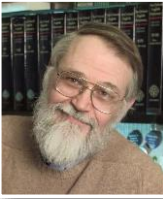


Bell Labs, Murray Hill (New Jersey), Foto: http://en.wikipedia.org/wiki/Bell_Labs

Bell Labs, New Jersey (USA), 1977

AWK: Aho, Weinberger, Kernighan

Introduction to Computing

Authors of AWK




Alfred Aho Peter Weinberger Brian Kernighan

<http://www.underforty.us/geeks.html>

Text processing (63)

Introduction to Computing

Fundamental question



What is text?

Introduction to Computing

Text

'I like PUT'

I		l	i	k	e		P	U	T
0	1	2	3	4	5	6	7	8	9

Text processing (65)

Introduction to Computing

A limerick

**It's a favourite project of mine,
A new value of π to assign.
I would fix it at 3,
For it's simpler, you see,
Than 3 point 1 4 1 5 9**

<http://www.freewebs.com/limericks/>
by Terry Walsh

Text processing (66)

Introduction to Computing

Input file

Jerzy	Nawrocki	43089	I1
Jane	Kowalski	43780	I2
Adam	Malinowski	43990	I1

Fields: \$1, \$2, \$3, ...

Text processing (67)

Introduction to Computing

Program structure

```
pattern1 {instruction1}
pattern2 {instruction2}
...      ...
```

Processing rule

Text processing (68)

Introduction to Computing

Execution principle

```
→ Jerzy Nawrocki
   Jane Kowalski
   Adam Malinowski

→ pattern1 {instruction1}
   pattern2 {instruction2}
   ...      ...
```

Text processing (69)

Introduction to Computing

Execution principle

```
→ Jerzy Nawrocki
   Jane Kowalski
   Adam Malinowski

→ pattern1 {instruction1}
   pattern2 {instruction2}
   ...      ...
```

Text processing (70)

Introduction to Computing

Execution principle

```
→ Jerzy Nawrocki
   Jane Kowalski
   Adam Malinowski

→ pattern1 {instruction1}
   pattern2 {instruction2}
   ...      ...
```

Text processing (71)

Introduction to Computing

Execution principle

```
→ Jerzy Nawrocki
   Jane Kowalski
   Adam Malinowski


→ pattern1 {instruction1}
   pattern2 {instruction2}
   ...      ...
```

\$0 denotes current line

Text processing (72)

Introduction to Computing

Agenda



- Simple programs
- Patterns
- Variables

Text processing (73)

Introduction to Computing

Simplest programs

How many lines on the output?

Jerzy	Nawrocki	43089	I1
Jane	Kowalski	43780	I2
Adam	Malinowski	43990	I1

```
$4=="I1" { print $2, $1; }
```

Nawrocki	Jerzy
Malinowski	Adam

Text processing (74)

Introduction to Computing

Simplest programs

Jerzy	Nawrocki	43089	I1
Jane	Kowalski	43780	I2
Adam	Malinowski	43990	I1

```
import sys
for line in sys.stdin:
    x= line.split()
    if x[3] == "I1":
        print(x[1], x[0])
```

\$1

\$4

Nawrocki	Jerzy
Malinowski	Adam

Text processing (75)

Introduction to Computing

Simplest programs

How many lines on the output?

Jerzy	Nawrocki	43089	I1
Jane	Kowalski	43780	I2
Adam	Malinowski	43990	I1

```
$4=="I1"
```

Jerzy	Nawrocki	43089	I1
Adam	Malinowski	43990	I1

Text processing (76)

Introduction to Computing

Simplest programs

How many lines on the output?

Jerzy	Nawrocki	43089	I1
Jane	Kowalski	43780	I2
Adam	Malinowski	43990	I1


```
{ print $2, $1; }
```

Nawrocki	Jerzy
Kowalski	Jane
Malinowski	Adam

Text processing (77)

Introduction to Computing

Agenda



- Simple programs
- Patterns
- Variables

Text processing (78)

Introduction to Computing

Execution principle

→ **BEGIN**

```
Jerzy Nawrocki
Jan Kowalski
EOF
```

→ pattern1 {instruction1}
pattern2 {instruction2}
... ..

Text processing (79)

Introduction to Computing

Execution principle

→ **BEGIN**

```
Jerzy Nawrocki
Jan Kowalski
EOF
```

→ pattern1 {instruction1}
pattern2 {instruction2}
... ..

Text processing (80)

Introduction to Computing

Execution principle

→ **BEGIN**

```
Jerzy Nawrocki
Jan Kowalski
EOF
```

→ pattern1 {instruction1}
pattern2 {instruction2}
... ..

Text processing (81)

Introduction to Computing

Execution principle

BEGIN

```
Jerzy Nawrocki
Jan Kowalski
```

→ **EOF**

→ pattern1 {instruction1}
pattern2 {instruction2}
... ..

Text processing (82)

Introduction to Computing

Beginning and end of text

```
Jerzy Nawrocki 43089 I1
Jane Kowalski 43780 I2
Adam Malinowski 43990 I1
```

↓

```
BEGIN { print "-----"; }
$4=="I2" { print $2, $1; }
END { print "*****"; }
```

↓

```
-----
Kowalski Jane
*****
```

Text processing (83)

Introduction to Computing

Beginning and end of text

```
Jerzy Nawrocki 43089 I1
Jan Kowalski 43780 I2
Adam Malinowski 43990 I1
```

↓

```
END { print "*****"; }
$4=="I2" { print $2, $1; }
BEGIN { print "-----"; }
```

↓

```
-----
Kowalski Jan
*****
```

Text processing (84)

Introduction to Computing

Relations

12	11
2	11

$\$1 > \2

12	11
----	----

Text processing (85)

Introduction to Computing

Compound patterns

\parallel or
 $\$1==1 \parallel \$2==1$

$\&\&$ and
 $\$1==1 \&\& \$2==1$

$!$ not
 $! \$1==1$

Text processing (86)

Introduction to Computing

Compound patterns

Jerzy	Adam	43089	I1
Adam	Kowalski	43780	I2
Adam	Malinowski	43990	I1


$\$4=="I1" \&\& \$1=="Adam" \{ \text{print } \$2, \$1; \}$

Malinowski	Adam
------------	------

Text processing (87)

Introduction to Computing

Agenda



- Simple programs
- Patterns
- **Variables**

Text processing (88)

Introduction to Computing

Variables

- Field variables $\$1, \$(i+j-1), ..$
- Variables introduced by a programmer
type: string of characters
initial value: empty string / zero
- Built-in variables
(standard meaning)

Text processing (89)

Introduction to Computing

Some built-in variables

NF – number of fields in a row
NR – row number
FILENAME – file name with input data

Text processing (90)

Introduction to Computing

Variables

NR	NF	total
1	2	0

→ If you have a hammer, everything looks like a nail

→ {total= total + NF;}
END {print "Fields: ", total;
print "Rows: ", NR;}

Text processing (91)

Introduction to Computing

Variables

NR	NF	total
1	2	0
		2

→ If you have a hammer, everything looks like a nail

→ {total= total + NF;}
END {print "Fields: ", total;
print "Rows: ", NR;}

Text processing (92)

Introduction to Computing

Variables

NR	NF	total
1	2	0
		2

→ If you have a hammer, everything looks like a nail

→ {total= total + NF;}
END {print "Fields: ", total;
print "Rows: ", NR;}

Text processing (93)

Introduction to Computing

Variables

NR	NF	total
1	2	0
2	1	2

→ If you have a hammer, everything looks like a nail

→ {total= total + NF;}
END {print "Fields: ", total;
print "Rows: ", NR;}

Text processing (94)

Introduction to Computing

Variables

NR	NF	total
1	2	0
2	1	2
		3

→ If you have a hammer, everything looks like a nail

→ {total= total + NF;}
END {print "Fields: ", total;
print "Rows: ", NR;}

Text processing (95)

Introduction to Computing

Variables

NR	NF	total
1	2	0
2	1	2
		3

→ If you have a hammer, everything looks like a nail

→ {total= total + NF;}
END {print "Fields: ", total;
print "Rows: ", NR;}

Text processing (96)

Introduction to Computing

Variables

NR	NF	total
1	2	0
2	1	2
		3

→ If you have a hammer, everything looks like a nail

→ {total= total + NF;}
END {print "Fields: ", total;
print "Rows: ", NR;}

Text processing (97)

Introduction to Computing

Variables

NR	NF	total
1	2	0
2	1	2
3	3	3

→ If you have a hammer, everything looks like a nail

→ {total= total + NF;}
END {print "Fields: ", total;
print "Rows: ", NR;}

Text processing (98)

Introduction to Computing

Variables

NR	NF	total
1	2	0
2	1	2
3	3	3
4	4	6
4	?	10

→ If you have a hammer, everything looks like a nail

→ {total= total + NF;}
END {print "Fields: ", total;
print "Rows: ", NR;}

Fields: 10
Rows: 4

Text processing (99)

Introduction to Computing

Pythonian version


If you have a hammer, everything looks like a nail

```
import sys
total=0
NR= 0
for line in sys.stdin:
    NR+= 1
    x= line.split()
    NF= len(x)
    total+= NF
print("Fields: " + str(total))
print("Rows: " + str(NR))
```

Fields: 10
Rows: 4

Introduction to Computing

Agenda




- Individual characters (C, Python)
- Strings of characters (C, Python)
- Rule-based text processing (AWK)
- Regular expressions (AWK, Python, C)

<http://digitalhistoryproject.com/the-medieval-scribe.html>

Text processing (101)

Introduction to Computing

Stephen Kleene



1909-01-05, Connecticut, USA
1934: Ph.D., Princeton Univ., (Alonzo Church)
1935: Univ. of Wisconsin-Madison (USA)
1939-40: Inst. for Advanced Study, Princeton – recursion theory
1990: National Medal of Sci.
1994-01-25, Madison

<http://www.math.wisc.edu/~gpslogic/>

Text processing (102)

Introduction to Computing

Regular expressions

Arithmetic expressions

Value: Text → Number

Value(**2**·**3** + **3**) = 9

Regular expressions

Value: Text → SetOfCharacterStrings

Value(**/Ala | Ola/**) = {"Ala", "Ola"}


Text processing (103)

Introduction to Computing

Regular expressions in Python

import re

res = re.search(regex, text)

 Check

Text processing (104)

Introduction to Computing

Example

It's a favourite project of mine,
A new value of Pi to assign.
I would fix it at 3,
For it's simpler, you see,
Than 3 point 1 4 1 5 9

<http://www.freewebs.com/limericks/>
by Terry Walsh


Select all the rows that contain 'ne'.

It's a favourite project of mine,
A new value of Pi to assign.

Introduction to Computing

Example

It's a favourite project of mine,
A new value of Pi to assign.
I would fix it at 3,
For it's simpler, you see,
Than 3 point 1 4 1 5 9



```
import sys
import re
for line in sys.stdin:
    if re.search("ne", line):
        print(line, end='')
```

It's a favourite project of mine,
A new value of Pi to assign.

Introduction to Computing

Regular expressions in C


```
#include <regex.h>
regex_t regRep;
char regex[], t[];
int er;
```

Compile

```
er = regcomp(&regRep, regex, cFlags)
```

Execute

```
er = regexec(&regRep, t, 0, NULL, eFlag)
regfree(&regRep)
```



Text processing (107)

Introduction to Computing

Example

It's a favourite project of mine,
A new value of Pi to assign.
I would fix it at 3,
For it's simpler, you see,
Than 3 point 1 4 1 5 9

<http://www.freewebs.com/limericks/>
by Terry Walsh

Select all the rows that contain 'ne'.

It's a favourite project of mine,
A new value of Pi to assign.

```
#include <stdio.h>
#include <stdlib.h>
#include <regex.h>
#define maxSize 100
int main(){
    regex_t regRep;
    char *regEx= "ne"; /* <-- Regular expression */
    char *txtPtr;
    size_t lineSize= maxSize-1;
    int er;
    txtPtr= (char *)malloc(maxSize);
    er= regcomp(&regRep, regEx, REG_EXTENDED | REG_NEWLINE);
    if (er != 0){
        printf("Error in regular expression\n");
        return 0; }
    while (getline(&txtPtr, &lineSize, stdin) > 0){
        er= regexec(&regRep, txtPtr, 0, NULL, 0);
        if (er == 0){
            printf("%s", txtPtr); /* <-- Action */
        }
    }
    regfree(&regRep);
    return 0; }
```

Introduction to Computing

Patterns with regular expressions

e.g. a character or string

\$0, \$1, \$2, ..

Whole string **String ~ /^ reg_exp \$/**

It's a favourite project of mine,
A new value of π to assign.
I would fix it at 3,
For it's simpler, you see,
Than 3 point 1 4 1 5 9


\$1 ~ /^I\$/

I would fix it at 3,

Introduction to Computing

Pythonian version

It's a favourite project of mine,
A new value of π to assign.
I would fix it at 3,
For it's simpler, you see,
Than 3 point 1 4 1 5 9

 **\$1**

```
import sys
import re
for line in sys.stdin:
    x= line.split()
    if re.search("^I$", x[0]):
        print(line, end='')

```

I would fix it at 3,

Introduction to Computing

Patterns with regular expressions

e.g. a character or string

\$0, \$1, \$2, ..

Whole string **String ~ /^ reg_exp \$/**

Beginning **String ~ /^ reg_exp /**

It's a favourite project of mine,
A new value of π to assign.
I would fix it at 3,
For it's simpler, you see,
Than 3 point 1 4 1 5 9

\$1 ~ /^I/

It's a favourite project of mine,
I would fix it at 3,

Introduction to Computing

Patterns with regular expressions

e.g. a character or string

\$0, \$1, \$2, ..

Whole string **String ~ /^ reg_exp \$/**

Beginning **String ~ /^ reg_exp /**

End **String ~ / reg_exp \$/**

It's a favourite project of mine,
A new value of π to assign.
I would fix it at 3,
For it's simpler, you see,
Than 3 point 1 4 1 5 9

\$3 ~ /e\$/

It's a favourite project of mine,
A new value of π to assign.

Introduction to Computing

Patterns with regular expressions

e.g. a character or string

\$0, \$1, \$2, ..

Whole string **String ~ /^ reg_exp \$/**

Beginning **String ~ /^ reg_exp /**

End **String ~ / reg_exp \$/**

Substring **String ~ / reg_exp /**

It's a favourite project of mine,
A new value of π to assign.
I would fix it at 3,
For it's simpler, you see,
Than 3 point 1 4 1 5 9

\$3 ~ /e/

It's a favourite project of mine,
A new value of π to assign.
For it's simpler, you see,

Introduction to Computing

Patterns with regular expressions

e.g. a character or string

\$0, \$1, \$2, ..

Whole string `String ~ /^ reg_exp $/`

Beginning `String ~ /^ reg_exp /`

End `String ~ / reg_exp $/`

Substring `String ~ / reg_exp /`

It's a favourite project of mine
A new value of π to assign.
I would fix it at 3,
For it's simpler, you see,
Than 3 point 1 4 1 5 9

\$0 ~ /ne/

It's a favourite project of mine,
A new value of π to assign.

Introduction to Computing

Patterns with regular expressions

e.g. a character or string

\$0, \$1, \$2, ..

Whole string `String ~ /^ reg_exp $/`

Beginning `String ~ /^ reg_exp /`

End `String ~ / reg_exp $/`

Substring `String ~ / reg_exp /`

It's a favourite project of mine,
A new value of π to assign.
I would fix it at 3,
For it's simpler, you see,
Than 3 point 1 4 1 5 9

/ne/

It's a favourite project of mine,
A new value of π to assign.

`$0 ~ /wyr_reg / = /wyr_reg /`

Introduction to Computing

Special characters

- . Any character
- [] Set of characters
- \n New line
- \. Dot
- \" Quotation
- \ddd Character of octal code = ddd

Text processing (117)

Introduction to Computing

Example

What will happen?

1
22
.
A
H2O

/^.\$/

1
.
A

Text processing (118)

Introduction to Computing

Agenda

- Individual characters
- Strings of characters
- Rule-based text processing
- Regular expressions

to be continued

Text processing (119)

Introduction to Computing

Bibliography

The **AWK** Programming Language

- A. Aho, B. Kernighan, P. Weinberger, *The AWK Programming Language*, Addison-Wesley, Reading, 1988.
- J. Nawrocki, W. Complak, Wprowadzenie do przetwarzania tekstów w języku AWK, *Pro Dialog 2* (1994), 23-46.
- J. Cybulka, B. Jankowska, J.R. Nawrocki, *Automatyczne przetwarzanie tekstów. AWK, Lex i YACC*, Nakom, Poznań, 2002.

Text processing (120)