Programming Language Design and Implementation (PLDI): CS-1319-1

Assignment-6

December 2023

1 Question 1

a) The global symbol contains the following:

```
int printStr(char *s);
int printInt(int n);
int readInt(int *eP);

void swap(int *a, int *b);
int kt(int n);
int main();
```

ST.glb				**Parent: Null**
Name	**Type**	**Category**	**Size**	**Offset**
printStr	ptr(char) → int	func	0	ST.printStr
printInt	int → int	func	0	ST.printInt
readInt	ptr(int) → int	func	0	ST.readInt
swap	void → int	func	0	ST.swap
kt	int*. int* → void	func	0	ST.kt
main	void → int	func	0	ST.main

b) Array of quad codes:

i) For function swap

```
100:  t = *a;
101:  *a = *b;
102:  *b = t;
104:  ret
```

ii) For function kt

```
offset _s1 "n"
100 :  t00 = n
101 :  p = n
102 :  t01 = 10
```

```
103 :  t02 = n % t01
104 :  d1 = t02
105 :  t03 = n / t01
106 :  n = t03
107 :  t04 = n % t01
108 :  d2 = t04
109 :  t05 = n / t01
110 :  n = t05
111:  if (d1 < d2) goto 113
112:  goto 118
113:  t06 = &d1
114:  param t06
115:  t07 = &d2
116 :  param t07
117:  call swap, 2
118:  t08 = n % t01
119:  d3 = t08
120:  if (d2 < d3) goto 122
121:  goto 134
122:  t09 = &d2
123:  param t10
124:  t10 = &d3
125:  param t10
126:  call swap, 2
127:  if (d1 < d2) goto 129
128:  goto 134
129:  t11 = &d1
130:  param t11
131:  t12 = &d2
132:  param t12
133:  call swap, 2
134:  param d1
135:  call printInt, 1
136:  param d2
137:  call printInt, 1
138:  param d3
139:  call printInt, 1
140:  param _s1
141:  call printStr, 1
142:  t13 = d1 - d3
143:  t14 = 90
144:  m = t13 * t14
145:  if m == p goto 147
```

```
146 :  goto 148
147:  ret m
148:  param m
149:  t15 = call kt, 1
150:  ret t15
```

iii) For the funtion main

```
offset _s1 "Constant = "
offset _s2 "\n"
100 :  t00 = 1
101 :  if t00 != 0 goto 103
102 :  goto 113
103 :  t01 = 0
104 :  param t01
105 :  call readInt, 1
106 :  param n
107 :  call kt, 1
108 :  param _s1
109 :  call printStr, 1
110 :  param m
111 :  call printInt, 1
112 :  param _s2
113 :  call printStr, 1
114 :  t02 = 0
115 :  ret t04
```

c)
Symbol table for function swap.

ST.swap				**Parent: ST.glb**
Name	**Type**	**Category**	**Size**	**Offset**
b	int*	param	4	+8
a	int*	param	4	+4
t	int	local	4	0

Symbol table of function main

ST.main				**Parent: ST.glb**
Name	**Type**	**Category**	**Size**	**Offset**
n	int	local	4	0
m	int	local	4	-4
t00...t02	int	local	4	-8...-16

Symbol table for function kt

ST.kt				Parent: ST.glb
Name	**Type**	**Category**	**Size**	**Offset**
n	int	param	4	+4
p	int	local	4	0
d1	int	local	4	-4
d2	int	local	4	-8
d3	int	local	4	-12
m	int	local	4	-16
t00...t15	int	temp	4	-20...-80

d)

2 Question 2

Peephole optimize the code of function kt :

```
offset _s1 "n"
 100 :  t00 = n ; deadcode
100 :  101 :  p = n
 102 :  t01 = 10 ; deadcode
101 :  103 :  t02 = n % t01
102 :  104 :  d1 = t02
103 :  105 :  t03 = n / t01
 106 :  n = t03 ; deadcode
104 :  107 :  t04 = n % t01
 108 :  d2 = t04 ; deadcode
105 :  109 :  t05 = n / t01
 110 :  n = t05 ; deadcode
106 :  111:  if (d1 >= d2) goto 118 (113);jump − over − jump : flippedd1 < d2
 112:  goto 118 ; fall through
107 :  113:  t06 = &d1
108 :  114:  param t06
109 :  115:  t07 = &d2
110 :  116 :  param t07
111:  117:  call swap, 2
112 :  118:  t08 = n % t01
113 :  119:  d3 = t08
114 :  120:  if (d2 >= d3) goto 134 (122) ; jump-over-jump:  flipped d2 < d3
 121:  goto 134 ; fall through
115 :  122:  t09 = &d2
116 :  123:  param t10
117 :  124:  t10 = &d3
```

```
118 :  125:  param t10
119 :  126:  call swap, 2
120 :  127:  if (d1 >= d2) goto 134 (129) ; jump-over-jump:  flipped d1 < d2
 128:  goto 134 ; fall through
121 :  129:  t11 = &d1
122 :  130:  param t11
123 :  131:  t12 = &d2
124 :  132:  param t12
125 :  133:  call swap, 2
126 :  134:  param d1
127 :  135:  call printInt, 1
128 :  136:  param d2
129 :  137:  call printInt, 1
130 :  138:  param d3
131 :  139:  call printInt, 1
132 :  140:  param _s1
133 :  141:  call printStr, 1
134 :  142:  t13 = d1 - d3
 143:  t14 = 90 ; deadcode
135 :  144:  m = t13 * t14
136 :  145:  if m != p goto 148 ; jump-over-jump :  flipped m == p
 146 :  goto 148 ; fall through
137 :  147:  ret m
138:  param m
139:  t15 = call kt, 1
140:  ret t15
```

3) Control Flow graph for fucntion kt:

i) Leader Quads
Leader Quads are chosen based on the following rules:
Rule 1 : First quad of the program
Rule 2 : quad's as target of some goto
Rule 3 : quad's following a conditional goto

```
offset _s1 "n"
100 :  p = n(Rule1)
101 :  t02 = n % t01
102 :  d1 = t02
103 :  t03 = n / t01
104 :  t04 = n % t01
105 :  t05 = n / t01
106 :  if (d1 >= d2) goto 118
107 :  t06 = &d1(Rule3)
108 :  param t06
```

```
109 :  t07 = &d2
110 :  param t07
111 :  call swap, 2
112 :  t08 = n % t01
113 :  d3 = t08
114 :  if (d2 >= d3) goto 134
115 :  t09 = &d2(Rule3)
116 :  param t10
117 :  t10 = &d3
118 :  param t10(Rule2)
119 :  call swap, 2
120 :  if (d1 >= d2) goto 134
121 :  t11 = &d1(Rule3)
122 :  param t11
123 :  t12 = &d2
124 :  param t12
125 :  call swap, 2
126 :  param d1
127 :  call printInt, 1
128 :  param d2
129 :  call printInt, 1
130 :  param d3
131 :  call printInt, 1
132 :  param _s1
133 :  call printStr, 1
134 :  t13 = d1 - d3(Rule2)
135 :  m = t13 * t14
136 :  if m != p goto 138
137 :  ret m(Rule3)
138 :  param m(Rule2)
139 :  t15 = call kt, 1
140 :  ret t15
```

ii) Basic block and the CFG :
Block B1:

```
100 :  p = n
101 :  t02 = n % t01
102 :  d1 = t02
103 :  t03 = n / t01
104 :  t04 = n % t01
105 :  t05 = n / t01
```

```
106 :  if (d1 >= d2) goto B4
 :  goto B2
```

Block B2:

```
107 :  t06 = &d1
108 :  param t06
109 :  t07 = &d2
110 :  param t07
111 :  call swap, 2
112 :  t08 = n % t01
113 :  d3 = t08
114 :  if (d2 >= d3) goto 134 B6
 :  goto B3
```

Block B3:

```
115 :  t09 = &d2
116 :  param t10
117 :  t10 = &d3
 :  goto B4
```

Block B4:

```
118 :  param t10
119 :  call swap, 2
120 :  if (d1 >= d2) goto B6
 :  goto B5
```

Block B5:

```
121 :  t11 = &d1
122 :  param t11
123 :  t12 = &d2
124 :  param t12
125 :  call swap, 2
126 :  param d1
127 :  call printInt, 1
```

```
128 :  param d2
129 :  call printInt, 1
130 :  param d3
131 :  call printInt, 1
132 :  param _s1
133 :  call printStr, 1
 :  goto B6
```

Block B6:

```
134 :  t13 = d1 - d3
135 :  m = t13 * t14
136 :  if m != p goto B8
 :  goto B7
```

Block B7:

```
137 :  ret m
```

Block B8:

```
138 :  param m
139 :  t15 = call kt, 1
140 :  ret t15
```